NAVSWC TR 91-487

AD-A242 322

SIGNIFICANT DIGIT COMPUTATION OF THE ELLIPSOIDAL COVERAGE FUNCTION AND ITS INVERSE

BY ARMIDO R. DIDONATO
STRATEGIC SYSTEMS DEPARTMENT

SEPTEMBER 1991

NAVAL SURFACE WARFARE CENTER
Dahlgren, Virginia 22448-5000 • Silver Spring, Maryland 20903-5000

SIGNIFICANT DIGIT COMPUTATION OF THE ELLIPSOIDAL COVERAGE FUNCTION AND ITS INVERSE

BY ARMIDO R. DIDONATO
STRATEGIC SYSTEMS DEPARTMENT

SEPTEMBER 1991

NAVAL SURFACE WARFARE CENTER
Dahlgren, Virginia 22448-5000 • Silver Spring, Maryland 20903-5000

FOREWORD

The work described in this report was performed in the Space and Surface Systems Division with partial support from the Computer and Information Systems Division of the Strategic Systems Department. Its purpose was to develop a new algorithm for the ellipsoidal coverage function, and to design associated Fortran software which is suitable for inclusion in a high quality mathematics and/or statistics subroutine library.

This document was administratively reviewed by J. L. Sloop, Head, Space and Surface Systems Division. The flowchart on page 13 was prepared by Dottie J. Burgess on a Macintosh IIx personal computer.

Approved by:

J. Ralph Fallin

J. Ralph Fallin, Acting Head
Strategic Systems Department

CONTENTS

I. INTRODUCTION

Three dimensional spherical coverage problems often appear in the study of weapon evaluations for aerial and submarine targets. Such studies require the capability to compute the ellipsoidal coverage function and its inverse. The ellipsoidal coverage function $P(\overline{R},\overline{H},\overline{K},\overline{L},u,v,w)$ represents the probability of an event occurring in $0x_1y_1z_1$-space within a sphere SP with radius $\overline{R}$ and center $(\overline{H},\overline{K},\overline{L})$, where

$$\text{SP:}\quad (x_1-\overline{H})^2+(y_1-\overline{K})^2+(z_1-\overline{L})^2=\overline{R}^{\,2}\,.$$

The random event occurs under an uncorrelated trivariate normal distribution with mean (0,0,0) and standard deviations u, v, w in the x_1, y_1, and z_1 directions, respectively. Thus

$$P(\overline{R},\overline{H},\overline{K},\overline{L},u,v,w)=\underset{\text{Volume of SP}}{\iiint}\mathfrak{F}(x_1,y_1,z_1,u,v,w)\,dx_1\,dy_1\,dz_1, \tag{1}$$

where

$$\mathfrak{F}(x_1,y_1,z_1,u,v,w)\equiv\frac{1}{(\sqrt{2\pi})^3\,u\,v\,w}E[x_1/(\sqrt{2}\,u)]\,E[y_1/(\sqrt{2}\,v)]\,E[z_1/(\sqrt{2}\,w)]\,,\qquad E(z)\equiv e^{-z^2}, \tag{2}$$

and

$$P(\overline{R},\overline{H},\overline{K},\overline{L},u,v,w)=P(\overline{R},\overline{K},\overline{H},\overline{L},v,u,w)=P(\overline{R},\overline{H},\overline{L},\overline{K},u,w,v)=P(\overline{R},|\overline{H}|,|\overline{K}|,|\overline{L}|,u,v,w). \tag{3}$$

Setting $x_1=\sqrt{2}\,u\,x$, $y_1=\sqrt{2}\,v\,y$, $z_1=\sqrt{2}\,w\,z$, one obtains

$$P=\frac{1}{\pi\sqrt{\pi}}\int_{L-R}^{L+R}E(z)\,dz\int_{K-Y}^{K+Y}E(y)\,dy\int_{H-X}^{H+X}E(x)\,dx\,, \tag{4}$$

where

$$X=\sqrt{\overline{R}^{\,2}-(\overline{L}-\sqrt{2}wz)^2-(\overline{K}-\sqrt{2}vy)^2}\Big/(\sqrt{2}\,u)\,,\qquad Y=\sqrt{\overline{R}^{\,2}-(\overline{L}-\sqrt{2}wz)^2}\Big/(\sqrt{2}\,v)\,,$$

and

$$\begin{cases} H=\overline{H}/(\sqrt{2}\,u)\,, & K=\overline{K}/(\sqrt{2}\,v)\,, & L=\overline{L}/(\sqrt{2}\,w)\\ R_1=\overline{R}/(\sqrt{2}\,u), & R_2=\overline{R}/(\sqrt{2}\,v), & R_3=\overline{R}/(\sqrt{2}\,w)=R. \end{cases} \tag{5}$$

It is easy to show, by using (4) with normalizations, that P is a function of six independent variables.

Rather than carry out the numerical triple integration of (4) directly we make use of an available computer program PKILL (or CIRCV) that yields the probability $P_c(\overline{r},\overline{H},\overline{K},u,v)$ of an event occurring under a bivariate normal distribution inside a circle in the $0x_1y_1$-plane, with center $(\overline{H},\overline{K})$ and radius $\overline{r}$. P_c is the two-dimensional analog of P [3, 4, 5, 6, 7].

Geometrically, one observes that P can be obtained by considering circular slices of SP parallel to the 0xy-plane. For a fixed z in [L − R, L + R], the xy-integration over a slice of radius $\overline{r}$ yields πP_c.

Weighting P_c with $E(z)/\sqrt{\pi}$ and integrating the result over z in $[L - R, L + R]$, gives P, i.e.,

$$P = \frac{1}{\sqrt{\pi}} \int_{L-R}^{L+R} E(z)\, P_c(\bar{r}, \overline{H}, \overline{K}, u, v)\, dz, \qquad \bar{r} = \sqrt{\overline{R}^2 - (\overline{L} - \sqrt{2}\, w z)^2}\,. \tag{6}$$

Another useful form for P can be obtained by splitting the integral in (6) into two pieces. One carries the integration from $L - R$ to L and the other from L to $L + R$. For the second let $z = 2L - t$, then combining the results gives

$$P = \frac{1}{\sqrt{\pi}} \int_{L-R}^{L} [E(t) + E(2L - t)]\, P_c(\bar{r}, \overline{H}, \overline{K}, u, v)\, dt, \qquad \bar{r} = \sqrt{\overline{R}^2 - (\overline{L} - \sqrt{2}\, w t)^2}\,. \tag{7}$$

The symmetry properties of P indicated by (1) and (3) allow $\overline{H}$, $\overline{K}$, $\overline{L}$ to be taken nonnegative. Since the integrand of P is positive and bounded, the order of integration in (1) is immaterial. Thus, as long as $\overline{H}$ is associated with u, $\overline{K}$ with v, and $\overline{L}$ with w, it does not matter which is called, say, $\overline{L}$ and w. For example, if the order of integration is chosen so that the original x-integration is performed last, then if initially $\overline{H} = 10$, $u = 5$, $\overline{L} = 20$, $w = 7$, we simply let $\overline{H} = 20$, $u = 7$, $\overline{L} = 10$, $w = 5$. In this way, we can refer to (6) or (7) as the basic representations for P, where $\overline{L}$ and w are always associated with the z-integration, with the understanding that the original order of integration may have been changed and the variables $\overline{H}$, $\overline{K}$, $\overline{L}$, u, v, w renamed as above.

The objective in this report is to expand on the work described in [8]. In [8] for $H^2 + K^2 + L^2 \le 10^{10}$ and $10^{-6} \le P \le .999999$, procedures were given for computing P or its inverse $\overline{R}$ (where P is given in place of $\overline{R}$) to 6 decimal-digit accuracy. Here procedures are given for computing P or $\overline{R}$ to 6 significant digits, when inherent error is negligible, for

$$H^2 + K^2 + L^2 \le 1/\text{eps}, \qquad 10^{-20} \le P \le .9999999, \tag{8}$$

where eps is the smallest positive machine dependent number such that $1 + \text{eps} > 1$. The double precision value of eps for the IBM PC is $2^{-52} \simeq 2.22 \cdot 10^{-16}$; in single precision eps$= 2^{-23} \simeq 1.19 \cdot 10^{-7}$. Moreover, for the smaller ranges used in [8], P or $\overline{R}$ can now be found to approximately 8 significant digits. All computations for testing were carried out in double precision PC Fortran on a Compaq Deskpro 386/20. The portable double precision Fortran function which yields P, given $\overline{R}$, $\overline{H}$, $\overline{K}$, $\overline{L}$, u, v, w, is called ELLCOV; the portable double precision subroutine which outputs $\overline{R}$, given P, $\overline{H}$, $\overline{K}$, $\overline{L}$, u, v, w, is called ELINV3. It is anticipated that the single precision version of these subprograms will be included in the Naval Surface Warfare Center (NAVSWC) Library of Mathematics Subroutines [13].

II. COMPUTATION OF P

Initially in the computation of P by ELLCOV four tests are used to eliminate most cases where $P \leq Z4 \equiv \max(10^{-50}, 100\ \text{epsm})$ or $P \geq 1 - E_1$, where epsm is the smallest machine dependent positive number the computer can use, and $E_1 \equiv \max(10^{-11},\ 50\ \text{eps})$. In single precision for the IBM PC, epsm $\simeq 1.17 \cdot 10^{-38}$; for double precision epsm $\simeq 2.22 \cdot 10^{-308}$.

Test #1 :

$$\boxed{\text{P set to 0} \quad \text{if} \quad \overline{R}^3 \leq 1.5\sqrt{2\pi}\ u\,v\,w\ Z4.}$$

For Test #2, starting from (1) with

$$\overline{D} \equiv \sqrt{\overline{H}^2 + \overline{K}^2 + \overline{L}^2}\,, \qquad M \equiv \max(u, v, w),$$

$$V = E[x_1/(\sqrt{2}\ u)]\ E[y_1/(\sqrt{2}\ v)]\ E[z_1/(\sqrt{2}\ w)],$$

we have, for $\overline{D} \geq \overline{R}$,

$$V \leq E[x_1/(\sqrt{2}\ M)]\ E[y_1/(\sqrt{2}\ M)]\ E[z_1/(\sqrt{2}\ M)] \leq E[(\overline{D} - \overline{R})/(\sqrt{2}\ \ M)].$$

Hence,

$$P \leq \frac{1}{2\pi\sqrt{2\pi}\ u\,v\,w}\ E[(\overline{D} - \overline{R})/(\sqrt{2}\ M)]\ \tfrac{4}{3}\pi\overline{R}^3 \leq Z4.$$

Test #2 :

$$\boxed{\begin{array}{c} \text{P set to 0} \quad \text{if} \quad \overline{R}^3\ E(\gamma) \leq 1.5\sqrt{2\pi}\ u\,v\,w\ Z4 \\ \gamma = (\overline{D} - \overline{R})/(M\ \sqrt{2}). \end{array}}$$

Test #3 :

$$\boxed{\begin{array}{c} \text{P set to 0} \quad \text{if} \quad \mathcal{T} > 9.6 \\ \mathcal{T} \equiv \max(H - R_1,\ \ K - R_2,\ \ L - R). \end{array}}$$

Test #4 :

$$\boxed{\begin{array}{c} \text{P set to 1} \quad \text{if} \quad P^* \equiv P(\overline{R}, \overline{H}, \overline{K}, \overline{L}, M, M, M) \geq 1 - E_1 \\ P^* \text{ is computed from (10) or (11).} \end{array}}$$

Generally, when none of the above tests are satisfied, P is computed by the numerical integration of (7). However, there are three situations where P can be evaluated without resorting to quadratures, and a fourth where P is given by P_c. We shall call these cases A, B, C, and D.

CASE A: $\boxed{\text{For small } \overline{R}}$

The equivalent of (1) is to take the normal distribution with mean at ($\overline{H}$, $\overline{K}$, $\overline{L}$) and to place the target sphere SP at the origin. Then using spherical coordinates

$$x_1 = \rho \cos\theta \sin\phi, \quad 0 \le \rho \le \overline{R}, \quad 0 \le \theta \le 2\pi, \quad 0 \le \phi \le \pi$$
$$y_1 = \rho \sin\theta \sin\phi$$
$$z_1 = \rho \cos\phi$$

with the volume element given by $dV = \rho^2 \sin\phi \, d\rho \, d\theta \, d\phi$, we obtain

$$P = \frac{1}{2\pi\sqrt{2\pi}\, u v w} \int_0^{\overline{R}} \int_0^{2\pi} \int_0^{\pi} E\left(\frac{\rho \cos\theta \sin\phi - \overline{H}}{\sqrt{2}\, u}\right) E\left(\frac{\rho \sin\theta \sin\phi - \overline{K}}{\sqrt{2}\, v}\right) E\left(\frac{\rho \cos\phi - \overline{L}}{\sqrt{2}\, w}\right) dV.$$

Expanding each of the exponentials about $\rho = 0$ and carrying out the integrations gives, after some tedious algebra,

$$P \simeq \frac{4}{3\sqrt{\pi}} R_1 R_2 R_3 \left(1 + \frac{1}{10} T + E_r\right) e^{-(H^2 + K^2 + L^2)}, \tag{9}$$

where H, K, L and R_1, R_2, R are defined in (5) with

$$T = 2\left[R_1{}^2(2H^2 - 1) + R_2{}^2(2K^2 - 1) + R_3{}^2(2L^2 - 1)\right]$$

$$E_r = \frac{1}{280}\left\{T^2 - 8\left[R_1{}^4(4H^2 - 1) + R_2{}^4(4K^2 - 1) + R_3{}^4(4L^2 - 1)\right]\right\}.$$

Then (9) is used to compute P when

$$\frac{1}{280}\left\{T^2 + 8 \left| R_1{}^4(4H^2 - 1) + R_2{}^4(4K^2 - 1) + R_3{}^4(4L^2 - 1)\right|\right\} \le \max\,[5 \cdot 10^{-8}, \text{eps}].$$

Summarized results for cases B and C follow. The derivations of the final expressions either have been given in referenced papers or they will be given in the Appendix A of this report.

CASE B: $\boxed{\text{For } u = v = w}$

$$P = \frac{1}{2}\,\text{aerf}(D, R) - \frac{2}{\sqrt{\pi}} R \left[\frac{1 - e^{-4RD}}{4RD}\right] E(D - R), \quad D \equiv \sqrt{H^2 + K^2 + L^2} \neq 0 \tag{10}$$

$$P = \text{erf}\, R - \frac{2}{\sqrt{\pi}} R\, E(R)\,, \qquad D = 0, \tag{11}$$

where E(z) is defined in (2), R and L are defined in (5), and

$$\text{aerf}(D, R) \equiv \text{erf}(D + R) - \text{erf}(D - R), \qquad \text{erf}\, x \equiv \frac{2}{\sqrt{\pi}} \int_0^x E(t)\, dt. \tag{12}$$

An algorithm is given in [9, App. A] which yields aerf to 13 significant digits. Equation(10) is derived in [8, App. A].

Subtraction of leading positive terms in (10) and (11) can result in loss of accuracy when computing P. Consider (11) for small R. Note that

$$\operatorname{erf} R = \frac{2}{\sqrt{\pi}} E(R) \sum_{0} \frac{2^n}{1 \cdot 3 \cdots (2n+1)} R^{2n+1}, \qquad [1, p.297]$$

so that the leading term cancels the 2nd term in (11). Therefore when $R < .071$, (11) is replaced by

$$P = \frac{2}{\sqrt{\pi}} E(R) \sum_{1} \frac{2^n}{3 \cdots (2n+1)} R^{2n+1}, \qquad D = 0. \tag{13}$$

In the case of (10) there are two situations where accuracy may be lost. First, consider $D - R$ large and $4RD > -\log eps$ (For the IBM PC, $\log eps \simeq -36.044$). Then with

$$\operatorname{aerf}(D, R) = \operatorname{erfc}(D - R) - \operatorname{erfc}(D + R), \qquad \operatorname{erfc} x = 1 - \operatorname{erf} x, \tag{14}$$

use

$$\operatorname{erfc} x \simeq \frac{E(x)}{\sqrt{\pi}\, x}\left[1 + \sum_{1} (-1)^n \frac{1 \cdot 3 \cdots (2n-1)}{(2x^2)^n}\right], \qquad (x \to \infty), \qquad [1, p.298] \tag{15}$$

in(14). It is easy to see that the term $\operatorname{erfc}(D + R)$ can be dropped since e^{-4RD} is negligible compared to one. Therefore (10) becomes

$$P \simeq \frac{E(D-R)}{2\sqrt{\pi}}\left[\frac{1}{D-R} + \frac{1}{D-R} \sum_{1} (-1)^n \frac{1 \cdot 3 \cdots (2n-1)}{[2(D-R)^2]^n}\right] - \frac{1}{2\sqrt{\pi}\, D} E(D-R),$$

or

$$P \simeq \frac{E(D-R)}{2\sqrt{\pi}\,(D-R)}\left[\frac{R}{D} + \sum_{1} (-1)^n \frac{1 \cdot 3 \cdots (2n-1)}{[2(D-R)^2]^n}\right], \qquad D \neq 0. \tag{16}$$

The relation (16) for P is used when $D - R > 4.25$, $4RD > -\log eps$, and $R > 0.425$.

Loss of digits can also occur when R is small and $D > 0$ in much the same way as was seen in obtaining (13). Here we use

$$P = \frac{4}{\sqrt{\pi}} E(D) \sum_{1} \frac{n\, H_{2n-1}(D)/D}{(2n+1)!} R^{2n+1}, \tag{17}$$

where $H_k(x)$ denotes the Hermite polynomial of degree k [1, p.771]. Equation (17) is used when $R < .425$. It is derived in Appendix A.

Subprogram ELLCOV calls subprogram EQSIG to evaluate P from the above relations when it recognizes Case B from its input.

CASE C: $\boxed{\text{For } u = v \text{ and } \overline{H} = \overline{K} = 0}$

Let

$$Z \equiv \sqrt{|1-(w/u)^2|}\,, \qquad S \equiv L - R Z^2, \qquad F \equiv L + R Z^2.$$

Using the fact, which is easily shown, that

$$P_c(\bar{r}, 0, 0, u, u) = 1 - E(r w/u), \qquad r = \sqrt{R^2 - (L-z)^2}, \qquad [5,6,7] \tag{18}$$

and substituting this result into (6) gives separate relations for P when u > w and w > u:

<u>u > w</u> $\qquad (i \equiv \sqrt{-1}\,)$

$$P = \tfrac{1}{2}\,\mathrm{aerf}(L,R) - \tfrac{1}{2Z}\,\exp\{-(\tfrac{w}{u})^2[R^2-(L/Z)^2]\}\;\mathrm{aerf}(\tfrac{L}{Z}, RZ)\,, \qquad L \neq 0, \quad S \leq 0 \tag{19}$$

$$P = \tfrac{1}{2}\,\mathrm{aerf}(L,R) - \frac{E(L-R)}{2Z}\Big[E(i\tfrac{S}{Z})\,\mathrm{erfc}(\tfrac{S}{Z}) - e^{-4RL}\,E(i\tfrac{F}{Z})\,\mathrm{erfc}(\tfrac{F}{Z})\Big], \quad L \neq 0, \quad S > 0 \tag{20}$$

$$P = \mathrm{erf}\,R - R\,E(Rw/u)\,\frac{\mathrm{erf}(RZ)}{RZ}, \quad L = 0. \tag{21}$$

<u>w > u</u>

$$P = \tfrac{1}{2}\,\mathrm{aerf}(L,R) - \frac{E(L-R)}{\sqrt{\pi}\,Z}\Big[\mathrm{daw}(\tfrac{F}{Z}) - e^{-4RL}\,\mathrm{daw}(\tfrac{S}{Z})\Big], \qquad L \neq 0 \tag{22}$$

$$P = \mathrm{erf}\,R - \tfrac{2}{\sqrt{\pi}}\,R\,E(R)\,\frac{\mathrm{daw}(RZ)}{RZ}, \quad L = 0\,, \tag{23}$$

where

$$\mathrm{daw}(x) \equiv E(x)\int_0^x e^{t^2}dt \qquad \text{(Dawson's integral)}. \qquad [1,\ \text{p.298}] \text{ and } [8] \tag{24}$$

Case C uses (19) – (23). They were derived in [8, pp. A4 – A7]. In a number of situations however these equations are written in different forms here to reduce the computational loss in accuracy. The modified forms appear below in (25) – (31) with their derivations given in Appendix A. In each of the modified relations given below the conditions under which they are used are specified first. The quantity P for case B is denoted here by P_B, and $H_n(x)$ denotes the Hermite polynomial of degree n. These results are used in the subroutine SEQHZ3 which is called by ELLCOV when the latter has recognized case C from the input. A flowchart for SEQHZ3 is given at the end of this section.

LET: $\quad \{\, 4RL \leq 10, \text{ and } R \leq \sqrt{2}, \text{ and } Rw/u \leq 1.$

$$P = \tfrac{2}{\sqrt{\pi}}\,R\,(Rw/u)^2\,E(L/\sqrt{2})\sum_1 F_{2n+1} \tag{25}$$

$$F_{2n+1} = \frac{1}{2n+1}\Big[G_{2n-1} - 2\,(Rw/u)^2\,F_{2n-1}\Big], \quad n \geq 1$$

$$G_{2n-1} = \frac{2R^{2n-2}}{(2n-1)(2n-2)!}\,E(L/\sqrt{2})\,H_{2n-2}(L)$$

$$F_1 = 0, \qquad H_0 = 1, \qquad G_1 = 2\,E(L/\sqrt{2})\,.$$

$$\text{LET:}\quad \left\{ \begin{array}{l} R > \sqrt{2}, \quad \text{or} \quad Rw/u > 1, \quad \text{and} \\ L = 0 \quad \text{and} \quad RZ \le \sqrt{3}. \end{array} \right.$$

$$P = P_B - \frac{2R}{\sqrt{\pi}} E(R) \sum_1 \frac{2^n}{3 \cdots (2n+1)} (RZ)^{2n}, \qquad u \ge w \tag{26}$$

$$P = P_B - \frac{2R}{\sqrt{\pi}} E(R) \sum_1 \frac{(-2)^n}{1 \cdot 3 \cdots (2n+1)} (RZ)^{2n}, \qquad w \ge u. \tag{27}$$

$$\text{LET:}\quad \left\{ \begin{array}{c} 4RL > 10 \quad \text{or} \quad R > \sqrt{2} \quad \text{or} \quad Rw/u > 1, \quad \text{and} \\ L \ne 0 \quad \text{or} \quad RZ > \sqrt{3}, \quad \text{and} \\ w/u > 1/10 \quad \text{or} \quad (w/u)^2 \max(R, L) > 1/2. \end{array} \right.$$

Let

$$\mathrm{erfcr}(x) \equiv 1/\sqrt{\pi} - x\, e^{x^2} \mathrm{erfc}\, x, \qquad x \ge 4$$

$$\mathrm{efsz} \equiv F\, \mathrm{erfcr}(S/Z) - S\, E(2\sqrt{RL})\, \mathrm{erfcr}(F/Z).$$

Then

$$P = P_B - \frac{E(R-L)}{2FS} \left\{ \frac{RZ^2}{\sqrt{\pi}} \left[4R^2Z^2 \frac{[1 - E(2\sqrt{RL})]}{4RL} + (1 + E(2\sqrt{RL}) \right] - \mathrm{efsz} \right\}, \qquad u > w, \quad \frac{S}{Z} \ge 5, \tag{28}$$

and

$$P = P_B - \frac{E(R-L)Z^2}{2\sqrt{\pi}FS} \left\{ 4R^3Z^2 \frac{[1 - E(2\sqrt{RL})]}{4RL} - R[1 + E(2\sqrt{RL})] + \mathrm{bdaw1} \right\}, \qquad w > u, \quad \frac{S}{Z} \ge 5, \tag{29}$$

where

$$\mathrm{daw}(y) \simeq 1/(2y)\, [1 + \mathrm{Frac}(y)/y^2], \qquad y \ge 5 \qquad [2]$$

$$\mathrm{bdaw1} = S/F^2\, \mathrm{Frac}(F/Z) - F/S^2\, E(2\sqrt{RL})\, \mathrm{Frac}(S/Z).$$

The computation of erfcr(x) is carried out by the Fortran function ERFCR given in [13]. The evaluation of daw(x) is based on minimax rational approximations [2]. For $t \ge 5$, the quantity Frac(t) represents a rational function in $1/t^2$. The function bdaw1 is computed by the Fortran subroutine BDAW1 given in Appendix B.

$$\text{LET:}\quad \left\{ \begin{array}{c} 4RL > 10 \quad \text{or} \quad R > \sqrt{2} \quad \text{or} \quad Rw/u > 1, \quad \text{and} \\ L = 0 \quad \text{or} \quad RZ > \sqrt{3}, \quad \text{and} \\ w/u > 1/10 \quad \text{or} \quad (w/u)^2 \max(R, L) > 1/2. \end{array} \right.$$

Let

$$\mathrm{dxdaw}(x) \equiv \mathrm{daw}(x)/x.$$

Then

$$P = 1/2\, \mathrm{aerf}(L, R) - (2/\sqrt{\pi})\, R\, E(L-R)\, \mathrm{dxdaw}(RZ), \qquad w > u, \quad L = 0, \quad \frac{S}{Z} < 5, \tag{30}$$

where dxdaw(x) is computed by the Fortran function DXDAW which is given in Appendix B.

$$\text{LET:} \quad \begin{cases} 4RL > 10 \quad \text{or} \quad R > \sqrt{2} \quad \text{or} \quad Rw/u > 1, \quad \text{and} \\ L \neq 0 \quad \text{or} \quad RZ > \sqrt{3}, \quad \text{and} \\ w/u \leq 1/10 \quad \text{or} \quad (w/u)^2 \max(R, L) \leq 1/2 \,. \end{cases}$$

With $i = \sqrt{-1}$,

$$P = \frac{1}{2}\left\{ \left(1 - e^{-(w/u)^2 [R^2 - L^2]}\right) \operatorname{aerf}(L, R) - e^{-(w/u)^2 [R^2 - L^2]} \sum_1 \frac{H_n(-i\,L w/u)}{n!} \frac{2}{\sqrt{\pi}} \int_{L-R}^{L+R} E(t)\, \left(\frac{-i\,w}{u}\right)^n t^n\, dt \right\}. \tag{31}$$

The recurrence relations used to generate (31) in SEQHZ3 are given in Appendix A.

CASE D: $\boxed{\text{For } L - R < -6 \text{ and } \bar{r} = \sqrt{\overline{R}^2 - (\overline{L} - \sqrt{2}\,w\,t)^2} \sim \overline{R}}$

$$P = P_c(\overline{R}, \overline{H}, \overline{K}, u, v). \tag{32}$$

This case is recognized when

$$(\overline{L} - \sqrt{2}\,w\,A)^2 \leq \overline{R}^2 \min(\text{eps}/3,\ 10^{-12}), \tag{33}$$

where A replaces the lower limit of integration in (7). For example, let $\overline{R} = 10^{10}$, $u = 10^{10}$, $v = 2$, $w = 1$, $\overline{H} = 10^{10}$, $\overline{K} = 1$, $\overline{L} = 2$. In this case $L - R = (2 - 10^{10})/\sqrt{2}$ can safely be replaced by $A = -7.0$ since $\bar{r}$ in (7) is an increasing function of t. Then

$$(\overline{L} - \sqrt{2}\,w\,A)^2/\overline{R}^2 \simeq 12 \cdot 10^{-20} < (2.22/3) \cdot 10^{-16}, \text{ and } \operatorname{erf}(7.0) \simeq 1 - 4 \cdot 10^{-23}.$$

Hence $P = P_c(\overline{R}, \overline{H}, \overline{K}, u, v) = .47724\ 986805$.

In general, excluding cases A,B,C,D, the probability P is computed from (7) by numerical integration. A 24th order Gaussian quadrature rule [1, p.916] is used for this purpose. The primary objectives are: (a) to determine the order of integration, i.e., which integration should be carried out last as the t-integration in (7), (b) to obtain effective limits of integration.

From Test#3, $\mathcal{T} \leq 9.60$. Indeed, if $L - R > 9.6$, then

$$P \leq \frac{1}{\sqrt{\pi}} \int_{9.6}^{\infty} E(z)\, P_c(\bar{r}, \overline{H}, \overline{K}, u, v)\, dz \leq \frac{1}{\sqrt{\pi}} \int_{9.6}^{\infty} E(z)\, dz = \frac{1}{2}\operatorname{erfc}(9.6) \simeq 2.8 \cdot 10^{-42}.$$

In addition, the lower limit in (6) or (7) can be restricted to values greater than -7.0. In fact, by Test#4 and (10), when $\overline{R} > \overline{D}$, then $P \geq P^* \simeq \operatorname{aerf}(D, R)/2$ where the second term in (10) can be neglected for $\overline{R} - \overline{D} > 7\sqrt{2}\,M$. Therefore since P is no less than P^*,

$$P > \operatorname{aerf}(D,R)/2 \geq \operatorname{erf}(7.0) \simeq 1 - 4 \cdot 10^{-23}.$$

Consequently, if we identify an initial effective integration range in (7) to be from A to B, then

$$A \equiv \max(L - R, -7.0), \qquad B \equiv \min(L, 9.6). \tag{34}$$

It is understood that an A and B are determined for the three different integration orders, i.e., with the original x,y, and the z-integrations reordered so that each is carried out last.

Let the integrand of (7) be denoted by G(t). Then ELLCOV calls TQUA1 to determine whether A can be raised and/or B lowered from the initial values given in (34). At each Gaussian abscissa t_i on [A,B] the integrand of (7), $G(t_i)$, is evaluated at increasing i starting at $i = 1$, where $A = t_0 < t_1 < \ldots < t_{23} < t_{24} < t_{25} = B$. Set the quantity $T8 = \max\,(100\text{ epsm}, 10^{-42})$, with epsm as defined on page 3. Then for the smallest t_i for which $G(t_i) > T8$, A is replaced by t_{i-1}. Similarly, starting at $j = 24$ and with decreasing j, the largest t_j for which $G(t_j) > T8$ is found. B is then replaced by t_{j+1}. The function G(t) is computed by the Fortran function FN2 given in Appendix B.

At this point three sets of integration limits A and B have been determined, one for each different last integration. Indicate these limits by AX, BX; AY, BY; AZ, BZ. The final order of integration is now selected as the one for which the integration interval $(B - A)$ is the largest. An exception to this choice is made if one lower limit of integration is < -2 and the other two lower limits are not negative, for example, $AX \leq -2$ and AY and AZ are not negative. In this particular case the x integration is chosen as the last integration. If the above exception does not hold and a tie occurs, say between the x and y integrations, $(BX - AX = BY - AY)$ the x-integration is carried out last if $AX > AY$ and $AX < -2$, otherwise the y-integration is carried out last. Choosing the order of integration in the ways described above is based on numerical studies and the heuristic argument that the more spike-like the integrand the more difficult it is to obtain an accurate numerical integration.

With the appropriate interchanges having been made, the integration to be carried out last has now been set and it is indicated by the z or t-integration as shown in (6) or (7), . Two further attempts are subsequently made to improve the values of A and B.

It may occur that for some $\bar{t}$ in [A,B], $P_c \simeq 1 - \epsilon$, $\epsilon = 2.3 \cdot 10^{-11}$, in which case, since $\bar{r}$ in (7) is an increasing function of t,

$$P \simeq \frac{1}{\sqrt{\pi}} \int_A^{\bar{t}} [E(t) + E(2L - t)]\, P_c(\bar{r}, \overline{H}, \overline{K}, u, v)\, dt + \frac{1}{2}\text{aerf}(L, \Upsilon),$$

where Υ is defined below. Such a situation is recognized by employing previous results from [3, p.15] and [4,7]. We have $P_c > 1 - 2.3 \cdot 10^{-11}$, for $t \geq \bar{t}$, if

$$\bar{r} = \sqrt{\overline{R}^2 - (\overline{L} - \sqrt{2}\,w\,t)^2} \geq \sqrt{\overline{H}^2 + \overline{K}^2} + 7m \equiv G, \qquad m \equiv \max(u, v),$$

or

$$|\bar{L} - \sqrt{2}\,w\,t\,| \leq \sqrt{\bar{R}^2 - G^2} \equiv \bar{\Upsilon},$$

or

$$\bar{t} = L - \Upsilon \leq t \leq L + \Upsilon, \qquad \Upsilon \equiv \bar{\Upsilon}/(\sqrt{2}\,w).$$

At this point a final effort is made to improve A and B. ELLCOV calls subroutine SQUAD which first evaluates the integrand in (7), G(t), at each Gaussian abscissa on [A, B] starting from the endpoints and moving symmetrically toward the center of [A, B]. An estimate is now available for P. With this estimate, the same procedure used in TQUA1 is carried out by SQUAD to possibly further improve A and/or B. The difference here is that the better estimate for P replaces T8 used in TQUA1. If however both A and B are unchanged by SQUAD then the value for P obtained from SQUAD gives the final result for P.

If A and/or B are improved by SQUAD, then ELLCOV calls subroutine RQUAD to obtain P by a final 24th order Gaussian numerical integration, based on the latest values of A and B found from SQUAD. In SQUAD or RQUAD, we have

$$P = \frac{1}{\sqrt{\pi}}\,\frac{B-A}{2}\sum_{i=12}^{1} y(i)\,[G(t_i^-) + G(t_i^+)],$$

where

$$G(t) = [E(t) + E(2L - t)]\; P_c[\,\bar{r}(t), \bar{H}, \bar{K}, u, v]$$

$$\bar{r}(t) = \sqrt{\bar{R}^2 - (\bar{L} - \sqrt{2}\,w\,t)^2}$$

$$t_i^- = \frac{B+A}{2} - x(i)\,\frac{B-A}{2}$$

$$t_i^+ = \frac{B+A}{2} + x(i)\,\frac{B-A}{2}\,.$$

y(i) = 24th order Gaussian weights on [− 1, 1]

x(i) = 24th order Gaussian abscissae on [− 1, 1]. [1, p. 916]

The x(i) and y(i) are stored in ELLCOV (see listing in Appendix B).

A parameter J is set in ELLCOV to either 0 or 1 or 2. It is set to 0 if, referring to P_c above, $\bar{H} = \bar{K} = 0$; it is set to 1 if u = v. In either of these two cases the subroutine CIRCV is used to evaluate P_c, otherwise the subroutine PKILL is used for this purpose with J set to 2. It is advantageous to use CIRCV if possible since it evaluates P_c more accurately than PKILL and is about ten times faster. However the order of the integration is determined, as described above, by TQUA1 and no effort is made to modify that choice.

The quantity $\bar{r}(t_i)$ can be simplified when A = L − R and = L. In this case, it is easy to show that

$$\begin{cases} \bar{r}(t_i^-) = \bar{R}\,U(i) = \bar{R}\sqrt{1-[1+x(i)]^2/4}\,, & i = 12, 11, \ldots, 2, 1 \\ \bar{r}(t_i^+) = \bar{R}\,U(12+i) = \bar{R}\sqrt{1-[1-x(i)]^2/4}\,, & i = 12, 11, \ldots, 2, 1. \end{cases}$$

The array U(j) has been precomputed and is available, thus eliminating the computation of 24 square roots and a possible loss of accuracy. When this feature comes into play, the parameter IL in SQUAD is set to 1. The array is stored in FN2 where G(t) is computed, (See the listing in Appendix B).

Although we claim 6 significant-digit accuracy for P, it may not be possible to obtain this accuracy for extreme values of the input where inherent error plays a dominant role. For example, using ELLCOV on an IBM PC with double precision Fortran (~ 16 significant digits), let

$$\bar{H} = 10^{13}, \quad \bar{K} = 0, \quad \bar{L} = 0, \quad u = 5, \quad v = w = 1,$$

then

$\bar{R}$	P
$1.0000\ 00000\ 0026 \cdot 10^{13}$	$.99999\ 9\underline{9004}$
$1.0000\ 00000\ 0025 \cdot 10^{13}$	$.99999\ 9\underline{7136}$
$1.0000\ 00000\ 0000 \cdot 10^{13}$	$.50000\ 00000$
$9.9999\ 99999\ 9999 \cdot 10^{12}$	$.49\underline{214\ 92357}$
$9.9999\ 99999\ 9600 \cdot 10^{12}$	$.\underline{62292\ 51932} \cdot 10^{-15}$
$9.9999\ 99999\ 9599 \cdot 10^{12}$	$.\underline{52932\ 23990} \cdot 10^{-15}$.

The underlined digits are in doubt.

A few more numerical results are included where inherent error is not a factor. If

$$\bar{H} = 1.0, \quad \bar{K} = 2.0, \quad \bar{L} = 3.0, \quad u = 2, \quad v = 4, \quad w = 1,$$

then

$\bar{R}$	P
$1.0000\ 00000\ 0000 \cdot 10^{-2}$	$.28764\ 95875 \cdot 10^{-9}$
$5.0000\ 00000\ 0000 \cdot 10^{-1}$	$.43140\ 77476 \cdot 10^{-4}$
1.0000 00000 0000	$.53583\ 15121 \cdot 10^{-3}$
3.0000 00000 0000	$.79694\ 49445 \cdot 10^{-1}$
4.0000 00000 0000	.25115 76137
5.0000 00000 0000	.47295 03432
7.0000 00000 0000	.78979 21674
$1.2000\ 00000\ 0000 \cdot 10^{1}$	.98945 55844
$1.8000\ 00000\ 0000 \cdot 10^{1}$	.99994 81467 .

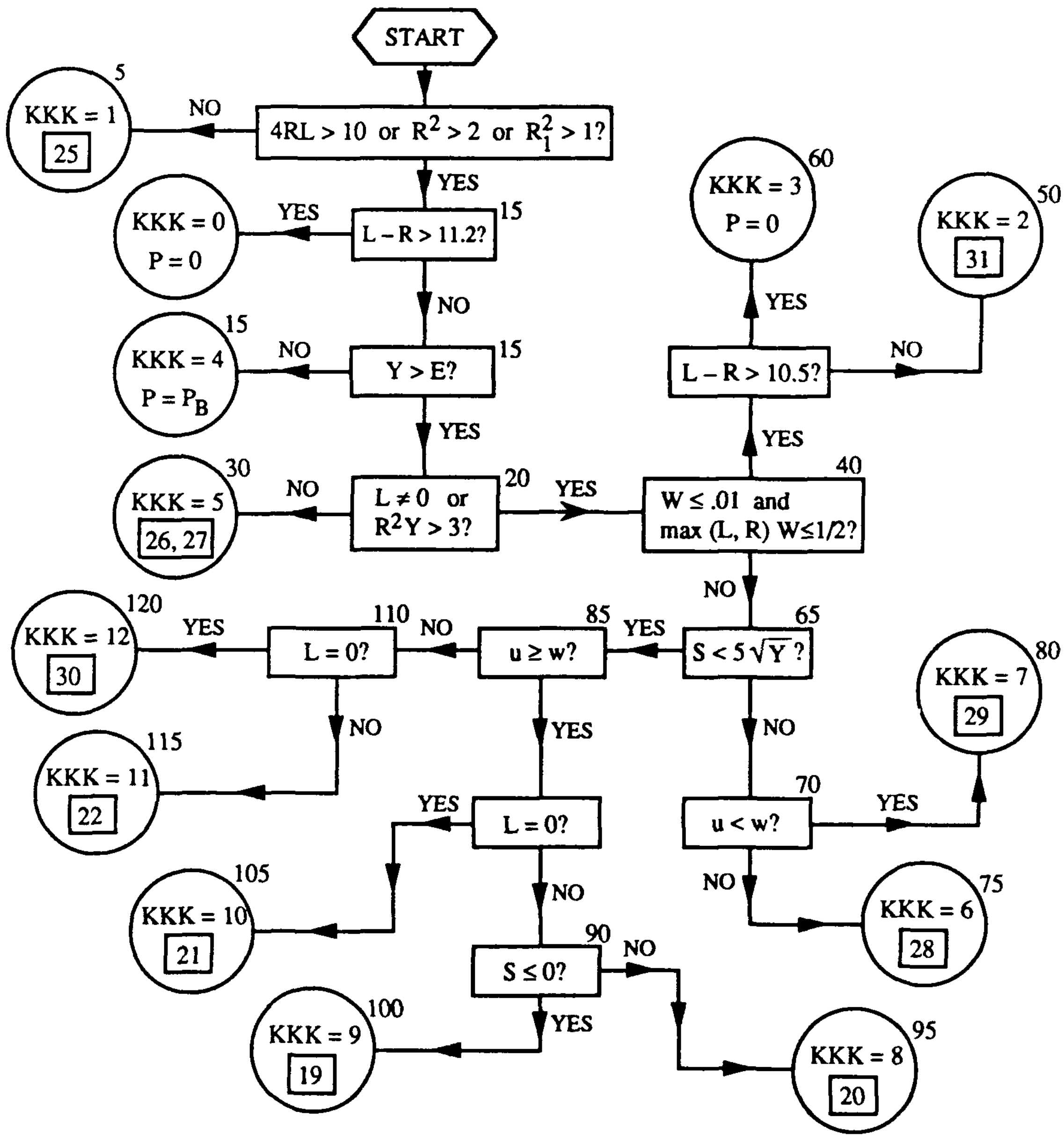
We also take the opportunity here to correct 3 typographical errors in Table 1 of [8, p.12].

$\overline{R} = 1$	$\overline{H} = 1/2$	$\overline{K} = \overline{L} = 0$	$u = v = w = 1$	$P = .17955\ 97978$
$\overline{R} = 1$	$\overline{H} = 2$	$\overline{K} = \overline{L} = 0$	$u = v = w = \sqrt{2/3}$	$P = .33473\ 93607 \cdot 10^{-1}$
$\overline{R} = 1$	$\overline{H} = 0$	$\overline{K} = \overline{L} = 0$	$u = 2v = 2w = 2/\sqrt{3}$	$P = .37539\ 30077$

Extensive testing was carried out to establish the accuracy claimed above for the computation of P by ELLCOV. This testing, using a Compaq Deskpro 386/20 PC, compared ELLCOV double precision results, with the results from an adaptive quadrature subroutine, DQAGS, which is contained in NSWCLIB [13].

FLOWCHART FOR SEQHZ3

CASE C: $H = K = 0,\ u = v \neq w,\ W \equiv (w/u)^2,\ Y \equiv |1 - W|,\ S \equiv L - RY$



• $P_B \equiv P$ for $u = v = w$.

• $E \equiv \max(\text{eps}/2, 10^{-10})$ See page 2 for eps.

• KKK: integer used to identify a particular branch of SEQHZ3.

• Numbers in rectangles refer to equation numbers of the text.

• Numbers outside boxes and circles refer to Fortran labels of the source code.

III. COMPUTATION OF $\overline{R}$ (The inverse problem)

In this section a procedure is described, given P, $\overline{H}$, $\overline{K}$, $\overline{L}$, u, v, w, to estimate $\overline{R}$. The method for estimating $\overline{R}$ within a given accuracy generally requires a sequence of values of P from ELLCOV. Since these computations usually involve time-consuming numerical triple integrations an effort is made to keep them to a minimum by obtaining a good early estimate of $\overline{R}$. Once such an estimate is found, halving, regula-falsi, and King's root finding procedure [12] are used to obtain a final estimate, $\widetilde{R}$, for $\overline{R}$. The objective is to obtain $\widetilde{R}$ to at least six significant digits (when inherent error does not contaminate the computations) for the range specified earlier in (8), namely

$$H^2 + K^2 + L^2 \le 1/\text{eps}, \qquad 10^{-20} \le P \le .9999999. \tag{35}$$

The Fortran subroutine ELINV3 yields $\widetilde{R}$. A listing of the code is given in Appendix C with listings for three additional portable supporting subprograms ELLRC, FCN1, SUB3.

The unknown value of $\overline{R}$ will always be contained in a known interval $[R_L, R_H]$. Initial values of the lower and upper bounds, R_L and R_H, are found for $\overline{R}$.

Let $I \equiv \max\,[3\sqrt{\pi/2}\ Puvw,\ (\sqrt{\pi/2}\ P\ M)^3]$, $M \equiv \max(u,v,w)$. Then

$$\overline{R} \ge R_L = \begin{cases} \overline{D}, \text{ if } P > 1/2 \text{ and } I \le \overline{D}^3, & \overline{D} \equiv \sqrt{\overline{H}^2 + \overline{K}^2 + \overline{L}^2} \\ I^{1/3}, \quad \text{otherwise.} & \end{cases} \tag{36}$$

The derivation of (36) is discussed in [8]. An initial value of R_H is obtained from the relation

$$\overline{R} \le R_H = \overline{D} + B(j)\,M \quad \text{with} \quad A(j) = P(B(j)M,0,0,0,M,M,M),$$

where B(j)M gives the radius of a sphere centered at the origin such that A(j) is the smallest value for which A(j) > P. This approximation is discussed more fully in [8; pp. 14-15]. The arrays A(k) and B(k) are given by

$A(1) = 10^{-30}$	$B(1) = 1.56 \cdot 10^{-10}$
$A(2) = 10^{-25}$	$B(2) = 7.23 \cdot 10^{-9}$
$A(3) = 10^{-20}$	$B(3) = 3.36 \cdot 10^{-7}$
$A(4) = 10^{-15}$	$B(4) = 1.56 \cdot 10^{-5}$
$A(5) = 10^{-10}$	$B(5) = 7.22 \cdot 10^{-4}$
$A(6) = 10^{-8}$	$B(6) = 3.36 \cdot 10^{-3}$
$A(7) = 5 \cdot 10^{-6}$	$B(7) = 2.66 \cdot 10^{-2}$
$A(8) = 10^{-4}$	$B(8) = 7.23 \cdot 10^{-2}$
$A(9) = 10^{-2}$	$B(9) = 0.339$
$A(10) = 0.10$	$B(10) = 0.765$
$A(11) = 0.30$	$B(11) = 1.1933$
$A(12) = 0.60$	$B(12) = 1.7170$
$A(13) = 0.90$	$B(13) = 2.5005$
$A(14) = 0.999$	$B(14) = 4.0335$
$A(15) = 0.999999$	$B(15) = 5.5380$
$A(16) = 0.99999999$	$B(16) = 6.3500$
$A(17) = 1.00$	$B(17) = 7.7000$

An improvement for $\overline{R}$ is generally obtained by using an estimate R_G given by F.E. Grubbs [11]. His estimate depends on the percentage point of the Chi-squared distribution which is determined by using the subroutine GAMINV contained in the NAVSWC math library [13]. The quantity actually given by Grubbs is $R_G{}^2$, namely

$$R_G{}^2 = \sigma^2\{[X(4/V_5, P) - 4/V_5]W_2 + Z\}, \tag{37}$$

where

$$\sigma^2 \equiv u^2 + v^2 + w^2$$

$$Z \equiv 1 + \overline{D}^2/\sigma^2, \qquad \overline{D}^2 \equiv \overline{H}^2 + \overline{K}^2 + \overline{L}^2$$

$$V_4 \equiv 2\left\{\frac{u^4}{\sigma^4}[1 + 4H^2] + \frac{v^4}{\sigma^4}[1 + 4K^2] + \frac{w^4}{\sigma^4}[1 + 4L^2]\right\}$$

$$T_5 \equiv 8\left\{\frac{u^6}{\sigma^6}[1 + 6H^2] + \frac{v^6}{\sigma^6}[1 + 6K^2] + \frac{w^6}{\sigma^6}[1 + 6L^2]\right\}$$

$$W_2 \equiv T_5/(2V_4), \qquad V_5 \equiv T_5{}^2/V_4{}^3,$$

and X(A,P) satisfies the incomplete gamma function relation

$$P = \frac{1}{\Gamma(\alpha)}\int_0^{X(\alpha,P)} e^{-t}\, t^{\alpha-1}\, dt, \qquad \alpha \equiv 4/V_5. \qquad [1, \text{p.}255]$$

The quantity $R_G{}^2$ from (37) occasionally may give a poor estimate for $\overline{R}^2$ or it may even give a negative value and consequently be of no use in such cases.

We attempt to improve our initial estimate for $\overline{R}$ by finding a constant $\overline{R}_C$, where

$$P = \int_{\overline{L}-\overline{R}_C}^{\overline{L}+\overline{R}_C}\int_{\overline{K}-\overline{R}_C}^{\overline{K}+\overline{R}_C}\int_{\overline{H}-\overline{R}_C}^{\overline{H}+\overline{R}_C} \mathfrak{F}(x_1, y_1, z_1, u, v, w)\, dx_1\, dy_1\, dz_1\,. \tag{38}$$

The quantity $\mathfrak{F}$ is defined in (2). The right hand side of (38) gives the cumulative normal probability P over the cube, with center at $(\overline{H}, \overline{K}, \overline{L})$ and equal sides parallel to the x_1, y_1, and z_1 axes, which contains the sphere with the same center, and of radius $\overline{R}_C$. The subroutine ELLRC, using Newton-Raphson, gives $\overline{R}_C$. Hence $\overline{R}_C < \overline{R} < \sqrt{3}\,\overline{R}_C$. The maximum number of Newton-Raphson iterations allowed by ELLRC is 40. The Fortran listing for ELLRC is given in Appendix C.

With the latest values for R_L and R_H and the corresponding P values P_L and P_H, a halving procedure is carried out until $P_L > 10^{-3}\, P_H$ and $(R_H - R_L) \le \overline{R}/10$. When both of the above inequalities are satisfied ELINV3 proceeds to obtain a final estimate for $\overline{R}$ by employing King's method [8, 12]. The method was described with a flow chart in [8]. It may be looked upon as a modified regula-falsi procedure. An estimate $\widetilde{R}$ for $\overline{R}$ is considered satisfactory if $|P(\widetilde{R}) - P| \le E8$, where E8 depends on P and is given by

$$E8 = \begin{cases} \max(10\text{ eps}, 10^{-8})\,P & 0 < P < .999 \\ \max(10\text{ eps}, 10^{-9}) & .999 \le P < .999999 \\ \max(10\text{ eps}, 10^{-10}) & .999999 \le P < .99999999 \\ \max(10\text{ eps}, 10^{-11}) & .99999999 \le P < 1.0. \end{cases} \quad \text{(See page 2 for eps).}$$

If the above inequality $|P(\widetilde{R}) - P| \le E8$ is satisfied an indicator IND is set to 0 in ELINV3. Other exits from ELINV3 are also identified by IND, namely,

IND = −1	$P < \max(10^{-40}, 10^{6}\text{ epsm})$	(See page 3 for epsm).
IND = +1	$P > 1 - \max(10^{-12}, 10\text{ eps})$	
IND = +2	$N \ge 30$	
IND = +3	$R_H - R_L < \max(10\text{ eps}, 10^{-14})\widetilde{R}$	
IND = +4	ELLCOV not able to yield $P(\widetilde{R})$ with the accuracy required.	

If IND = $|1|$, P has been specified outside allowable limits. The output $\widetilde{R}$ is set to -10^{10}. Let N denote the number of iterations. If IND = 2, then the maximum allowable number of iterations (30) for finding $\widetilde{R}$ was reached. N is increased by one for each call to ELLCOV for which its output is nonzero. Extensive checking has found $N \le 27$. If IND = 3, then more digits are required in the $\overline{R}$ estimates than are available, .i.e, the word length for the particular machine in use is too short. If IND = 4, then the subprogram that yields $P(\widetilde{R})$, ELLCOV, cannot produce the accuracy required. This difficulty can only appear if P is very near one and it is due to either the limitations of the numerical integration or to inherent error.

Some numerical results from ELINV3 are given below in Table 1 (4 parts) which reproduces Table 2 of [8] with improved estimates for $\overline{R}$. The notation for Table 1 follows. Let $\overline{P}$ denote the input values of P and let $\widetilde{R}$ denote the final estimate for $\overline{R}$ using ELINV3. RERR represents the relative error given by $|P(\widetilde{R}) - \overline{P}|/\overline{P}$. IND and N have been defined above.

TABLE 1 (PART 1)

$u = 2.0$	$v = 4.0$		$w = 6.0$	
$\overline{H} = 5.00$	$\overline{K} = 10.0$		$\overline{L} = 20.0$	
$\widetilde{R}$	RERR	IND	N	$\overline{P}$
$0.3183274539 \cdot 10^{1}$	$+0.260 \cdot 10^{-8}$	0	8	$0.500000 \cdot 10^{-5}$
$0.1028906156 \cdot 10^{2}$	$-0.392 \cdot 10^{-13}$	0	8	$0.500000 \cdot 10^{-2}$
$0.1660907596 \cdot 10^{2}$	$-0.641 \cdot 10^{-12}$	0	6	0.100000
$0.2339931591 \cdot 10^{2}$	$+0.163 \cdot 10^{-9}$	0	5	0.500000
$0.3048621773 \cdot 10^{2}$	$+0.708 \cdot 10^{-8}$	0	7	0.900000
$0.4077123752 \cdot 10^{2}$	$-0.231 \cdot 10^{-10}$	0	7	0.999000
$0.4844201564 \cdot 10^{2}$	$+0.956 \cdot 10^{-10}$	0	6	0.999995

TABLE 1 (PART 2)

$u = 5.0 \cdot 10^{-5}$	$v = 5.0 \cdot 10^{-2}$		$w = 1.0$	
$\overline{H} = 0.20$	$\overline{K} = 1.00$		$\overline{L} = 5.00$	
$\widetilde{R}$	RERR	IND	N	$\overline{P}$
$0.1171110995 \cdot 10^{1}$	$+0.653 \cdot 10^{-9}$	0	9	$0.500000 \cdot 10^{-5}$
$0.2629994892 \cdot 10^{1}$	$-0.127 \cdot 10^{-12}$	0	8	$0.500000 \cdot 10^{-2}$
$0.3855992354 \cdot 10^{1}$	$-0.143 \cdot 10^{-11}$	0	6	0.100000
$0.5103195068 \cdot 10^{1}$	$+0.287 \cdot 10^{-12}$	0	7	0.500000
$0.6364036803 \cdot 10^{1}$	$+0.572 \cdot 10^{-8}$	0	7	0.900000
$0.8154468511 \cdot 10^{1}$	$-0.259 \cdot 10^{-10}$	0	7	0.999000
$0.9472449895 \cdot 10^{1}$	$+0.919 \cdot 10^{-9}$	0	7	0.999995

TABLE 1 (PART 3)

	$u = 2.0$		$w = 1.0$	
	$\overline{H} = 0.00$	$v = 5.0$	$\overline{L} = 10.0$	
		$\overline{K} = 5.00$		
$\widetilde{R}$	RERR	IND	N	$\overline{P}$
$0.6184610273 \cdot 10^{1}$	$-0.780 \cdot 10^{-9}$	0	10	$0.500000 \cdot 10^{-5}$
$0.8185500434 \cdot 10^{1}$	$-0.900 \cdot 10^{-10}$	0	13	$0.500000 \cdot 10^{-2}$
$0.9733490650 \cdot 10^{1}$	$-0.119 \cdot 10^{-12}$	0	6	0.100000
$0.1172659410 \cdot 10^{2}$	$+0.211 \cdot 10^{-10}$	0	6	0.500000
$0.1545659369 \cdot 10^{2}$	$+0.910 \cdot 10^{-8}$	0	6	0.900000
$0.2295500292 \cdot 10^{2}$	$+0.285 \cdot 10^{-12}$	0	8	0.999000
$0.2902552953 \cdot 10^{2}$	$+0.241 \cdot 10^{-9}$	0	6	0.999995

TABLE 1 (PART 4)

	$u = 2.0 \cdot 10^{4}$	$v = 2.0 \cdot 10^{9}$	$w = 1.0$	
	$\overline{H} = 0.0$	$\overline{K} = 2.0 \cdot 10^{5}$	$\overline{L} = 0.0$	
$\widetilde{R}$	RERR	IND	N	$\overline{P}$
$0.2137608898 \cdot 10^{5}$	$-0.379 \cdot 10^{-8}$	0	6	$0.500000 \cdot 10^{-5}$
$0.1253323942 \cdot 10^{8}$	$-0.374 \cdot 10^{-8}$	0	1	$0.500000 \cdot 10^{-2}$
$0.2513226958 \cdot 10^{9}$	$-0.580 \cdot 10^{-9}$	0	1	0.100000
$0.1348979507 \cdot 10^{10}$	$+0.942 \cdot 10^{-10}$	0	1	0.500000
$0.3289707270 \cdot 10^{10}$	$+0.697 \cdot 10^{-11}$	0	1	0.900000
$0.6581053496 \cdot 10^{10}$	$+0.541 \cdot 10^{-13}$	0	1	0.999000
$0.9129575506 \cdot 10^{10}$	$+0.222 \cdot 10^{-15}$	0	1	0.999995

IV. TEST RESULTS FOR ELLCOV AND ELINV3

Extensive testing was carried out to estabiish the accuracy claimed for ELLCOV and ELINV3 keeping in mind that inherent error plays a major role when D is large (see page 11). The tests compared ELLCOV results with the corresponding outputs from a double precision adaptive quadrature subroutine, DQAGS, which is contained in NSWCLIB [13]. For ELINV3, values of P ranging from 10^{-20} to .9999999 were given with extensive ranges of the variables $\overline{H}$, $\overline{K}$, $\overline{L}$, u, v (w = 1). An estimate $\widetilde{R}$ for the true value $\overline{R}$ was found using ELINV3. ELLCOV was then used with $\widetilde{R}$ to compute $\widetilde{P}$ for comparison with the initial given value of P.

Testing using a Compaq Deskpro 386/20 computer with a CYRIX numeric coprocessor showed that the average time to compute a value of P using ELLCOV was ~ 0.4 seconds. The average time using ELINV3 to find a value $\widetilde{R}$ was ~ 1.1 seconds requiring on the average approximately 6 iterations. The maximum observed number of iterations, 27, occurred for isolated cases in the range $10^{-20} \leq P \leq 10^{-3}$. The maximum observed time for ELINV3 was 25 seconds, also occurring for small values of P. All testing was done using double precision Fortran for the IBM PC.

REFERENCES

1. Abramowitz, M. and Stegun, I. (EDS.), *Handbook of Mathematical Functions,* Dover, New York, N.Y. 1965.
2. Cody, W. J., Paciorek, K. A., and Thacher, H. C., Chebyshev Approximations to Dawson's Integral, *Math. of Comp.*, 24, 1970, pp. 171 – 178.
3. DiDonato, A. R. and Jarnagin, M. P., *Integration of the General Bivariate Gaussian Distribution over an Offset Ellipse,* NWL Report 1710, Aug. 1960, Naval Weapons Laboratory, Dahlgren, Va. 22448.
4. DiDonato, A. R. and Jarnagin, M. P., Integration of the General Bivariate Gaussian Distribution over an Offset Circle, *Math. of Comp.*, 15, #76, Oct. 1961, pp. 375-382.
5. DiDonato, A. R. and Jarnagin, M.P., *A Method for Computing the Generalized Circular Error Function and the Circular Coverage Function,* NWL Report No. 1768 23, Jan. 1962, Naval Weapons Laboratory, Dahlgren, Va. 22448.
6. DiDonato, A. R. and Jarnagin, M.P., A Method for Computing the Circular Coverage Function , *Math. of Comp.*, July, 1962, pp. 347 – 355.
7. DiDonato, A. R., *Five Statistical Programs in Basic for Desktop Computers,* Report NSWC TR 83-13, Nov. 1982, Naval Surface Weapons Center, Dahlgren, Va. 22448.
8. DiDonato, A. R., *Integration of the Trivariate Normal Distribution over an Offset Sphere and an Inverse Problem,* Report NSWC TR 87-27, Naval Surface Warfare Center, Feb. 1988, Dahlgren, Va. 22448.
9. DiDonato, A. R., *Significant Digit Computation of the Elliptical Coverage Function,* Report NAVSWC TR 90 – 513, Sept. 1990, Naval Surface Warfare Center, Dahlgren, Va. 22448.
10. DiDonato, A. R. and Morris, A. H., Computation of the Incomplete Gamma Function Ratios and Their Inverse, *ACM Trans. on Math. Software,* 12, #4, Dec. 1986, pp. 377-393.
11. Grubbs, F. E., Approximate Circular and Noncircular Offset Probabilities of Hitting, *Op. Res.*, 12, 1964, pp. 51-62.
12. King, R.F., An Improved Pegasus Method for Root Finding, *BIT,* 13, 1973, pp.423-427.
13. Morris, A. H., Jr., *NSWC Library of Mathematical Subroutines,* Report NSWC TR 90-21, Jan. 1990, Naval Surface Warfare Center, Dahlgren, Va. 22448.

APPENDIX A

DERIVATION OF (17) AND (25) – (31)

DERIVATION OF (17) AND (25) – (31)

DERIVATION OF (17):

Equation (17) holds when u = v = w and D ≠ 0 and is given by

$$P = \frac{4}{\sqrt{\pi}} E(D) \sum_{1} \frac{n\, H_{2n-1}(D)/D}{(2n+1)!} R^{2n+1}, \qquad D \equiv \sqrt{H^2 + K^2 + L^2}, \tag{A(1)}$$

where $H_k(x)$ denotes the classical Hermite polynomial of degree k. From (10)

$$P = \frac{1}{2}\left\{ \mathrm{erf}(D+R) - \mathrm{erf}(D-R) - \frac{1}{\sqrt{\pi}\, D}[E(D-R) - E(D+R)] \right\}. \tag{A(2)}$$

From [9, p. A-3]

$$\mathrm{aerf}(D,R) \equiv \mathrm{erf}(D+R) - \mathrm{erf}(D-R) = \frac{4}{\sqrt{\pi}} E(D) \sum_{n=0}^{\infty} \frac{H_{2n}(D)\, R^{2n+1}}{(2n+1)!}. \tag{A(3)}$$

Using the property of the Hermite polynomials

$$\frac{d^n}{dx^n} E(x) = (-1)^n E(x)\, H_n(x), \qquad [1, 22.11.7] \tag{A(4)}$$

and expanding E(D + R) about R = 0 gives

$$E(D+R) = E(D) \sum_{n=0}^{\infty} (-1)^n \frac{H_n(D)\, R^n}{n!}. \tag{A(5)}$$

With these results A(2) becomes

$$P = \frac{1}{2}\left\{ \frac{4}{\sqrt{\pi}} E(D) \sum_{n=0}^{\infty} \frac{H_{2n}(D)\, R^{2n+1}}{(2n+1)!} - \frac{2}{\sqrt{\pi}} \frac{E(D)}{D} \sum_{n=0}^{\infty} \frac{H_{2n+1}(D)\, R^{2n+1}}{(2n+1)!} \right\}, \tag{A(6)}$$

or

$$P = \frac{1}{\sqrt{\pi}} \frac{E(D)}{D} \sum_{n=0}^{\infty} \frac{R^{2n+1}}{(2n+1)!} \left[2D\, H_{2n}(D) - H_{2n+1}(D) \right]. \tag{A(7)}$$

Making use of the recurrence relation for Hermite polynomials, namely,

$$H_{n+1}(x) = 2x\, H_n(x) - 2n\, H_{n-1}(x), \qquad H_0 = 1, \quad H_1 = 2x, \quad [1, 22.7.13] \tag{A(8)}$$

in A(7) yields A(1).

DERIVATION OF (25):

Equation (25) is given by

$$P = \frac{2}{\sqrt{\pi}} R\, (Rw/u)^2\, E(L/\sqrt{2}) \sum_{1} F_{2n+1} \tag{A(9)}$$

$$F_{2n+1} = \frac{1}{2n+1}\left[G_{2n-1} - 2\,(Rw/u)^2\, F_{2n-1} \right], \quad n \geq 1 \tag{A(9.1)}$$

$$G_{2n-1} = \frac{2\, R^{2n-2}}{(2n-1)\,(2n-2)!} E(L/\sqrt{2})\, H_{2n-2}(L) \tag{A(9.2)}$$

$$F_1 = 0, \qquad H_0 = 1, \qquad G_1 = 2\, E(L/\sqrt{2}). \tag{A(9.3)}$$

Let

$$f(R) \equiv \int_{L-R}^{L+R} E(z) \left\{1 - e^{-(w/u)^2[R^2-(L-z)^2]}\right\} dz. \qquad A(10)$$

Then, for $H = K = 0$ and $u = v$, we have from (18) and (6)

$$P = \frac{1}{\sqrt{\pi}} f(R). \qquad A(11)$$

Our aim is to obtain the Taylor series for $f(R)$ about $R = 0$ and also the recurrence relation for computing the successive terms of the series. From A(10) and (12)

$$f(R) = \frac{\sqrt{\pi}}{2} \operatorname{aerf}(L,R) - \int_{L-R}^{L+R} E(z)\, e^{-(w/u)^2[R^2-(L-z)^2]} dz. \qquad A(12)$$

Note that $f(R)$ is an odd function of R so that $f^{(2n)}(0) = 0$, $n = 0,1,2,\ldots$, where

$$f^{(n)}(R) \equiv \frac{d^n}{dR^n} f(R).$$

Thus

$$f(R) = \sum_{n=0}^{\infty} \frac{f^{(2n+1)}(0)\, R^{2n+1}}{(2n+1)!}. \qquad A(13)$$

Differentiating $f(R)$ in A(12) gives

$$\frac{1}{2}\left(\frac{u}{w}\right)^2 f^{(1)}(R) = R \int_{L-R}^{L+R} E(z)\, e^{-(w/u)^2[R^2-(L-z)^2]} dz$$

$$= R\,[Q(R) - f(R)], \quad f^{(1)}(0) = 0, \qquad A(14)$$

where, using A(3) with $D = L$, we have introduced the notation

$$Q(R) \equiv \frac{\sqrt{\pi}}{2} \operatorname{aerf}(L,R) = 2\,E(L) \sum_{n=0}^{\infty} \frac{H_{2n}(L)\, R^{2n+1}}{(2n+1)!} \quad \text{[See A(3)]}. \qquad A(15)$$

Successive differentiations beginning with A(14) give

$$\frac{1}{2}\left(\frac{u}{w}\right)^2 f^{(2)}(R) = Q(R) - f(R) + R\,[Q^{(1)}(R) - f^{(1)}(R)]$$

$$\frac{1}{2}\left(\frac{u}{w}\right)^2 f^{(3)}(R) = 2\,[Q^{(1)}(R) - f^{(1)}(R)] + R\,[Q^{(2)}(R) - f^{(2)}(R)]$$

$$\frac{1}{2}\left(\frac{u}{w}\right)^2 f^{(4)}(R) = 3\,[Q^{(2)}(R) - f^{(2)}(R)] + R\,[Q^{(3)}(R) - f^{(3)}(R)]$$

$$\vdots$$

$$\frac{1}{2}\left(\frac{u}{w}\right)^2 f^{(n)}(R) = (n-1)\,[Q^{(n-2)}(R) - f^{(n-2)}(R)] + R\,[Q^{(n-1)}(R) - f^{(n-1)}(R)].$$

Hence, it follows that

$$\frac{1}{2}\left(\frac{u}{w}\right)^2 f^{(2n+1)}(0) = 2n\,[Q^{(2n-1)}(0) - f^{(2n-1)}(0)], \qquad A(16)$$

where, from A(15),

$$Q^{(2n-1)}(0) = 2\, E(L)\, H_{2n-2}(L), \qquad Q^{(1)}(0) = 2\, E(L).$$

Now define F_{2n+1} by

$$2\,R\,(R\tfrac{w}{u})^2\, E(L\sqrt{2})\, F_{2n+1} = \frac{f^{(2n+1)}(0)}{(2n+1)!}\, R^{2n+1}, \qquad n = 1,2,\ldots \tag{A(16)}$$

Then, from A(11), A(13), and A(16)

$$P = \frac{2}{\sqrt{\pi}} R\,(R\tfrac{w}{u})^2\, E(L/\sqrt{2}) \sum_{n=1}^{\infty} F_{2n+1}\,, \qquad F_1 = 0, \tag{A(17)}$$

where, using A(15) and A(16),

$$F_{2n+1} = 2n\left[\frac{Q^{(2n-1)}(0)}{(2n-1)!} - 2(\tfrac{w}{u})^2\, \frac{\frac{1}{2}(\frac{u}{w})^2 f^{(2n-1)}(0)}{(2n-1)!}\right]\frac{R^{2n-2}}{2n\,(2n+1)}$$

$$= \frac{1}{2n+1}\left[G_{2n-1} - 2(R\tfrac{w}{u})^2\, F_{2n-1}\right], \qquad n \geq 1, \tag{A(17.1)}$$

and

$$G_{2n-1} = \frac{2\,E(L/\sqrt{2})\, R^{2n-2}}{(2n-1)!}\, H_{2n-2}(L). \tag{A(17.2)}$$

Also

$$F_1 = 0, \qquad H_0(L) = 1, \qquad G_1 = 2\,E(L/\sqrt{2}), \tag{A(17.3)}$$

where the first relation follows from setting $n = 0$ in A(16) and using A(14).

DERIVATION OF (26) – (27):

When $H = K = L = 0$ and $u = v$, one obtains from (21) for $u > w$

$$P = \operatorname{erf} R - R\,E(R\,w/u)\,\frac{\operatorname{erf}(RZ)}{RZ}, \qquad Z \equiv \sqrt{|1-(w/u)^2|}\,. \tag{A(18)}$$

Then using

$$\operatorname{erf}(RZ) = \frac{2}{\sqrt{\pi}}\, E(RZ) \sum_{0} \frac{2^n}{1\cdot 3\cdots(2n+1)}\,(RZ)^{2n+1}, \qquad [1,\ \text{p.}297]$$

in A(18) yields (26),

$$P = P_B - \frac{2R}{\sqrt{\pi}} E(R) \sum_{1} \frac{2^n}{3\cdots(2n+1)}\,(RZ)^{2n}, \qquad u \geq w,$$

where

$$P_B = \operatorname{erf} R - \frac{2}{\sqrt{\pi}}\, R\, E(R) \qquad [\text{See (11)}].$$

For $w > u$, one obtains from (23)

$$P = \operatorname{erf} R - \frac{2\,R}{\sqrt{\pi}}\, E(R)\, \frac{\operatorname{daw}(RZ)}{RZ}, \quad L = 0\,, \tag{A(19)}$$

where

$$\operatorname{daw}(x) \equiv E(x) \int_0^x e^{t^2} dt \qquad \text{(Dawson's integral).} \quad [1,\ \text{p.}298\ ] \text{ and } [8]$$

Also

$$\mathrm{daw}(x) = \int_0^x e^{(t^2 - x^2)}\, dt = x \int_0^{\pi/2} E(x\cos\theta)\cos\theta\, d\theta, \quad t = x\sin\theta$$

$$= \sum_0 (-1)^n \frac{x^{2n+1}}{n!} \int_0^{\pi/2} \cos^{2n+1}\theta\, d\theta$$

$$= \sum_0 (-1)^n \frac{x^{2n+1}}{n!} \frac{2^n\, n!}{1\cdot 3\cdot 5 \cdots 2n+1}.$$

Using this result, with $x = RZ$, in A(19) above gives (27),

$$P = P_B - \frac{2R}{\sqrt{\pi}} E(R) \sum_1 \frac{(-2)^n}{1\cdot 3 \cdots (2n+1)} (RZ)^{2n}, \qquad w \geq u.$$

DERIVATION OF (28) – (29):

For $H = K = 0$ and $u = v$, with $S \geq 5Z$, we derive the following relations:

$$P = P_B - \frac{E(R-L)}{2FS}\left\{\frac{RZ^2}{\sqrt{\pi}}\left[4R^2Z^2\frac{[1 - E(2\sqrt{RL})]}{4RL} + [1 + e^{-4RL}]\right] - \mathrm{efsz}\right\}, \quad u > w \qquad \text{A(20)}$$

$$\mathrm{erfcr}\, x \equiv 1/\sqrt{\pi} - x\, e^{x^2} \mathrm{erfc}\, x, \qquad x \geq 4 \qquad \text{A(20.1)}$$

$$\mathrm{efsz} \equiv F\, \mathrm{erfcr}(S/Z) - S\, e^{-4RL}\, \mathrm{erfcr}(F/Z), \qquad \text{A(20.2)}$$

and

$$P = P_B - \frac{E(R-L)}{2FS}\left\{\frac{RZ^2}{\sqrt{\pi}}\left[4R^2Z^2\frac{[1 - E(2\sqrt{RL})]}{4RL} - [1 + e^{-4RL}]\right] + \frac{Z^2}{\sqrt{\pi}}\mathrm{bdaw1}\right\}, \quad w > u \qquad \text{A(21)}$$

$$\mathrm{daw}(y) \simeq 1/(2y)\,[1 + \mathrm{Frac}(y)/y^2], \qquad y \geq 5 \qquad [2] \qquad \text{A(21.1)}$$

$$\mathrm{bdaw1} = S/F^2\, \mathrm{Frac}(F/Z) - F/S^2\, E(2\sqrt{RL})\, \mathrm{Frac}(S/Z), \qquad \text{A(21.2)}$$

with

$$Z \equiv \sqrt{|1 - (w/u)^2|}, \qquad S \equiv L - RZ^2, \qquad F \equiv L + RZ^2. \qquad \text{A(22)}$$

For the derivation of (28), which is A(20), first consider (20), with $u > w$,

$$P = \frac{1}{2}\,\mathrm{aerf}(L,R) - \frac{E(L-R)}{2Z}\left[E(i\tfrac{S}{Z})\,\mathrm{erfc}(\tfrac{S}{Z}) - e^{-4RL}\, E(i\tfrac{F}{Z})\,\mathrm{erfc}(\tfrac{F}{Z})\right], \quad L \neq 0, \quad S > 0. \quad \text{A(23)}$$

The quantity in square brackets in A(23) can now be written, using A(20.1) and A(20.2), as

$$\frac{Z[F - S\, e^{-4RL}]}{\sqrt{\pi}FS} - \frac{Z}{FS}\mathrm{efsz}.$$

Substituting this result into A(23) and also adding and subtracting

$$\frac{2}{\sqrt{\pi}} R \left[\frac{1 - e^{-4RL}}{4RL}\right] E(L-R) \frac{2(L^2 - R^2 Z^4)}{2FS}, \qquad A(24)$$

gives, after using A(22),

$$P = P_B - \frac{E(L-R)}{2FS} \left\{ 4RL^2 \frac{[1 - e^{-4RL}]}{\sqrt{\pi} 4RL} + \frac{RZ^2}{\sqrt{\pi}} [1 + e^{-4RL}] - \text{efsz} \right\} +$$

$$2R \frac{[1 - e^{-4RL}]}{\sqrt{\pi} 4RL} E(L-R) \frac{2(L^2 - R^2 Z^4)}{2FS}, \qquad A(25)$$

where P_B is given by (10). Carrying out the obvious cancellation in A(25), followed by some minor rearrangements gives A(20).

In order to derive (29) above begin with (22), where $w > u$, .i.e.,

$$P = \frac{1}{2} \text{aerf}(L,R) - \frac{E(L-R)}{\sqrt{\pi} Z} \left[\text{daw}\left(\frac{F}{Z}\right) - e^{-4RL} \text{daw}\left(\frac{S}{Z}\right) \right], \quad L \neq 0. \qquad A(26)$$

The quantity in square brackets in A(26) can now be written, using A(21.1) and A(21.2), as

$$\frac{Z}{2FS} \left\{ S - F e^{-4RL} + Z^2 \left[\frac{S}{F^2} \text{Frac}(F/Z) - e^{-4RL} \frac{F}{S^2} \text{Frac}(S/Z) \right] \right\}.$$

Substituting this result into A(26) and adding and subtracting A(24) gives, using A(22),

$$P = P_B - \frac{E(L-R)}{2FS\sqrt{\pi}} \left\{ 4RL^2 \frac{[1 - e^{-4RL}]}{4RL} - RZ^2 [1 + e^{-4RL}] + Z^2 \text{ bdaw1} \right\} +$$

$$2R \frac{[1 - e^{-4RL}]}{\sqrt{\pi} 4RL} E(L-R) \frac{2(L^2 - R^2 Z^4)}{2FS}. \qquad A(27)$$

Carrying out the obvious cancellation in A(27), followed by some minor rearrangements, gives A(21) or (29).

DERIVATION OF (30):

Equation (30) follows directly from (23), where $L = 0$, by noting that $\text{aerf}(L,R) = 2 \,\text{erf} R$, by multiplying and dividing by R after the minus sign, and by using $\text{dxdaw} \equiv \text{daw}(x)/x$.

DERIVATION OF (31):

For the derivation of (31) use A(10) and A(11) to obtain

$$P = \frac{1}{\sqrt{\pi}} \int_{L-R}^{L+R} E(z) \left\{ 1 - e^{-(w/u)^2 [R^2 - L^2]} e^{-(w/u)^2 [2Lz - z^2]} \right\} dz. \qquad A(28)$$

Let

$$\mathcal{Y} \equiv e^{-(w/u)^2 [2Lz - z^2]},$$

and make the substitution $z = i \frac{u}{w} \xi$ $(i \equiv \sqrt{-1})$ to obtain

$$\mathcal{I} \equiv e^{2(-i\frac{w}{u}L)\xi - \xi^2}. \qquad A(29)$$

Using the relation

$$e^{2xt - t^2} = \sum_0 \frac{H_n(x)\,t^n}{n!}, \qquad [1, p. 784]$$

and A(29), A(28) becomes

$$P = \frac{1}{\sqrt{\pi}} \int_{L-R}^{L+R} E(z) \left\{ 1 - e^{-(w/u)^2 [R^2 - L^2]} \sum_0 \frac{H_n(-i\frac{w}{u}L)(\frac{w}{iu}z)^n}{n!} \right\} dz. \qquad A(30)$$

With obvious modifications to A(30) the result for (31) follows, namely,

$$P = \frac{1}{2}\left\{ \left(1 - e^{-(w/u)^2 [R^2 - L^2]}\right) \mathrm{aerf}(L, R) \right.$$

$$\left. - e^{-(w/u)^2 [R^2 - L^2]} \sum_1 \frac{H_n(-iLw/u)}{n!} \frac{2}{\sqrt{\pi}} \int_{L-R}^{L+R} E(t)\, (\frac{-iw}{u})^n\, t^n dt \right\}. \qquad A(31)$$

Equation A(31) is evaluated with the use of recurrence relations. Let

$$T_n \equiv \frac{2}{\sqrt{\pi}} \int_{L-R}^{L+R} E(t)\, t^n dt \Big/ n!$$

$$= \frac{(L-R)^{n-1} E(L-R) - (L+R)^{n-1} E(L+R)}{\sqrt{\pi}\, n!} + \frac{1}{2n} T_{n-2}$$

$$= \frac{1}{2n} T_{n-2} + \frac{1}{\sqrt{\pi}} \frac{E(L-R)}{n!} \left[(L-R)^{n-1} - e^{-4LR} (L+R)^{n-1} \right], \quad n \ge 2,$$

where

$$T_0 = \mathrm{aerf}(L, R), \qquad T_1 = \frac{1}{\sqrt{\pi}} E(L-R)\,[1 - e^{-4RL}].$$

Let

$$U_n \equiv (\tfrac{w}{u})^n (-i)^n H_n(-i\tfrac{w}{u}L). \qquad A(32)$$

Then using the recurrence relation for Hermite polynomials given in A(8), we have

$$U_n = \left[(-i)\tfrac{2w}{u}(-i)\tfrac{w}{u}L\right] U_{n-1} + 2(n-1)(\tfrac{w}{u})^2 U_{n-2}, \qquad n \ge 2,$$

or

$$U_n = 2(\tfrac{w}{u})^2 \left[-L\,U_{n-1} + (n-1)\,U_{n-2} \right], \qquad n \ge 2,$$

with

$$U_0 = 1, \qquad U_1 = -2(\tfrac{w}{u})^2 L.$$

Therefore A(31) can also be written as

$$P = \frac{1}{2}\left\{ \left(1 - e^{-(w/u)^2 [R^2 - L^2]}\right) \mathrm{aerf}(L, R) - e^{-(w/u)^2 [R^2 - L^2]} \sum_1 T_n\, U_n \right\}. \qquad A(33)$$

APPENDIX B

FORTRAN LISTINGS FOR ELLCOV AND SUPPORTING ROUTINES

FORTRAN LISTINGS FOR ELLCOV AND SUPPORTING ROUTINES

Below we give a summary of the various subprograms that are used for computing P. These subprograms are listed in this appendix, except for those which are contained in NSWCLIB [13]. The NSWCLIB routines are identified below with a superscript *. All subprograms are given in double precision.

REFERENCING OF ROUTINES USED TO COMPUTE P

ELLCOV	uses:	AERF*	CH	CIRCV*	DEPSLN*	DPMPAR*	EQSIG
		PKILL*	RQUAD	SEQHZ3	SQUAD	TQUA1	
EQSIG	uses:	AERF*	DXPARG*	ERF5	HSEXP		
FN2	uses:	CIRCV*	PKILL*				
RQUAD	uses:	FN2					
SEQHZ3	uses:	AERF*	BDAW1	DAW*	DXDAW	DXPARG*	EQSIG
		ERF*	ERFC0*	ERFCR	HSEXP*		
SQUAD	uses:	FN2					
TQUA1	uses:	FN2					

```
      DOUBLE PRECISION FUNCTION ELLCOV(R,HX,HY,HZ,SX,SY,SZ,
     *     XK0,XK2,SS)
C-----------------------------------------------------------------
C     (X,Y,Z) IS AN ELEMENT OF A CARTESIAN COORDINATE SYSTEM.
C     ELLCOV RETURNS THE PROBABILITY OF A SHOT FALLING, UNDER AN
C     ELLIPSOIDAL NORMAL DISTRIBUTION, IN A SPHERE OF RADIUS R WITH
C     CENTER (HX,HY,HZ) AND RADIUS R. THE DISTRIBUTION HAS MEAN
C     (0,0,0) AND STANDARD DEVIATIONS SX, SY AND SZ IN THE X,Y,Z
C     DIRECTIONS, RESPECTIVELY. THE INPUT PARAMETERS ARE R,HX,HY,HZ,
C     SX,SY,SZ,XK0,XK2,SS, WHERE XK0 = HX*HX+HY*HY+HZ*HZ,
C     XK2 = SQRT(XK0) AND SS = MAX(SX,SY,SZ). A 24TH ORDER GAUSSIAN
C     NUMERICAL INTEGRATION IS CARRIED OUT IN RQUAD AND SQUAD.
C
C     THE OUTPUT ELLCOV IS ACCURATE TO AT LEAST 6-SIGNIFICANT DIGITS
C     WHEN 1D-20 .LT. ELLCOV .LT. .9999999, AND (H/SX)**2+(K/SY)**2+
C     (L/SZ)**2 .LT. 2/DPMPAR(1).
C
C     REF: INTEGRATION OF THE TRIVARIATE NORMAL DISTRIBUTION OVER AN
C       OFFSET SPHERE AND AN INVERSE PROBLEM. NSWC TR. 87-27, 2/1988.
C     REF: SIGNIFICANT DIGIT COMPUTATION OF THE ELLIPSOIDAL COVERAGE
C       FUNCTION AND ITS INVERSE. NAVSWC TR. 91-487, 8/91.
C-----------------------------------------------------------------------------
      EXTERNAL             AERF  ,CH    ,DEPSLN ,DPMPAR ,EQSIG  ,PKILL
      EXTERNAL             RQUAD ,SEQHZ3 ,SQUAD ,TQUA1  ,CIRCV
C-----------------------------------------------------------------------------
      DOUBLE PRECISION     AERF  ,DEPSLN    ,DPMPAR    ,EQSIG
      DOUBLE PRECISION     X(12) ,Y(12)
      DOUBLE PRECISION     K     ,L
C-----------------------------------------------------------------------------
      DOUBLE PRECISION     A     ,AH    ,AK    ,AL    ,A2    ,A3
      DOUBLE PRECISION     B     ,BH    ,BK    ,BL    ,B1    ,C5
      DOUBLE PRECISION     C6    ,C7    ,C8    ,C9    ,D     ,DEP
      DOUBLE PRECISION     EPS   ,E1    ,FH    ,FK    ,FL    ,F2
      DOUBLE PRECISION     H     ,HH    ,HK    ,HL    ,HX    ,HY
      DOUBLE PRECISION     HZ    ,PY    ,R     ,R1    ,R2    ,R3
      DOUBLE PRECISION     SS    ,SX    ,SY    ,SZ    ,S1    ,S2
      DOUBLE PRECISION     S3    ,T5    ,T8    ,W1    ,XK0   ,XK2
      DOUBLE PRECISION     XK2   ,XK7   ,ZH    ,ZL    ,Z4
C-----------------------------------------------------------------
C    X(*),Y(*) -- GAUSSIAN ABSCISSAS AND WEIGHTS OF ORDER 24 ON (-1,1).
C-----------------------------------------------------------------
       DATA X(1) /6.40568928626056D-2/ ,X(2) /.191118867473616D0/,
     *      X(3) /.315042679696163D0/  ,X(4) /.433793507626045D0/,
     *      X(5) /.545421471388840D0/  ,X(6) /.648093651936976D0/,
     *      X(7) /.740124191578554D0/  ,X(8) /.820001985973903D0/,
     *      X(9) /.886415527004401D0/  ,X(10) /.938274552002733D0/,
     *      X(11) /.974728555971309D0/ ,X(12) /.995187219997021D0/
C-----------------------------------------------------------------
```

```
      DATA  Y(1)  /.127938195346752D0/  ,Y(2)  /.125837456346828D0/,
     *      Y(3)  /.121670472927803D0/  ,Y(4)  /.115505668053726D0/,
     *      Y(5)  /.107444270115966D0/  ,Y(6)  /9.76186521041139D-2/,
     *      Y(7)  /8.61901615319533D-2/ ,Y(8)  /7.33464814110803D-2/,
     *      Y(9)  /5.92985849154368D-2/ ,Y(10) /4.42774388174198D-2/,
     *      Y(11) /2.85313886289337D-2/ ,Y(12) /1.23412297999872D-2/
C-----------------------------------------------------------------
      DATA B1 /0.5641895835477563D0/,C5 /7.0D0/,
     *   C6 /0.3989422804014327D0/,C7 /0.7071067811865475D0/,
     *   C8 /1.414213562373095D0/ ,C9 /9.6D0/, F2 /1D1/
C-----------------------------------------
C       B1 = 1/SQRT(PI),     C6 = 1/SQRT(2PI)
C-----------------------------------------
      EPS = 10*DPMPAR(1)
      T8 = 1D2*DPMPAR(2)
      Z4 = MAX(T8,1D-25*1D-25)
      DEP = -DEPSLN(0)
      N = 12
      J1 = 0
      H = ABS(HX)
      K = ABS(HY)
      L = ABS(HZ)
      S1 = SX
      S2 = SY
      S3 = SZ
      PY = 0.D0
      ELLCOV = 0.0D0
      E1 = 1.5D0*SX*SY*SZ*Z4/C6
      XK7 = R*R*R
      IF (XK7 .LE. E1) RETURN
      A3 = (XK2 - R)*C7/SS
      A = EXP(-A3*A3)
      IF (XK2 .LT. R) GO TO 5
      IF (XK7*A .LE. E1) RETURN
      GO TO 10
C-------------------------------
    5 PY = EQSIG(R,XK0,XK2,SS,EPS,DEP,A)
      E1 = DMAX1(1D-11, 5*EPS)
      IF (PY .LT. 1.0D0 - E1) GO TO 10
      ELLCOV = 1.0D0
      RETURN
   10 IF (SX .NE. SY .OR. SY .NE. SZ) GOTO 15
C---------------------------------------------
C            S1 = S2 = S3.
C---------------------------------------------
      ELLCOV = PY
      IF (R .GT. XK2) RETURN
      ELLCOV = EQSIG(R,XK0,XK2,SS,EPS,DEP,A)
      RETURN
```

```
C-----------------------
C       SMALL R
C-----------------------
 15   R1 = R/S1
      R2 = R/S2
      R3 = R/S3
      HH = H/S1
      HK = K/S2
      HL = L/S3
      AH = R1*R1
      BH = HH*HH
      AK = R2*R2
      BK = HK*HK
      AL = R3*R3
      BL = HL*HL
      T5 = AH*(BH - 1D0) + AK*(BK - 1D0) + AL*(BL - 1D0)
      IF (ABS(T5) .GT. 1D-3) GOTO 20
      W1 = 4*(AH*AH*(BH - .5D0) + AK*AK*(BK - .5D0) +
     *      AL*AL*(BL - .5D0))
      AH =  (T5*T5 + ABS(W1))/280
      IF (AH .GT. MAX(EPS,2.5D-8)) GOTO 20
      ELLCOV = B1*R1*R2*R3/(3*C7)*(1.D0 + T5/10 + (T5*T5 - W1)/280)*
     *            EXP(-(BH + BK + BL)/2)
      RETURN
 20   J1 = 3
      IF (SX .EQ. SY .AND. H + K .EQ. 0.D0) GOTO 25
      J1 = 2
      IF (SX .EQ. SZ .AND. H + L .EQ. 0.D0) GOTO 25
      J1 = 1
      IF (SZ .NE. SY .OR. L + K .NE. 0.D0) GOTO 30
 25   CALL CH(J1,H,K,L,S1,S2,S3)
C-------------------------------------
C        S1 = S2 AND H = K = 0.
C-------------------------------------
      CALL SEQHZ3(R,L,S1,S3,EPS,DEP,XK0,A,PY,ELLCOV,KKK)
      RETURN
C-------------------------------------
 30   J = 2
      II = 0
      AH = MAX(-C5,(HH - R1)*C7)
      IF (AH .GT. C9) RETURN
      BH = MIN(F2,HH*C7)
      AK = MAX(-C5,(HK - R2)*C7)
      IF (AK .GT. C9) RETURN
      BK = MIN(F2,HK*C7)
      AL = MAX(-C5,(HL - R3)*C7)
      IF (AL .GT. C9) RETURN
      BL = MIN(F2,HL*C7)
```

```
      T8 = MAX(1D-42,T8)
      ZL = MIN(EPS/30,1D-12)
      CALL TQUA1(AH,BH,N,X,R,K,L,H,S2,S3,S1,PY,T8,DEP,FH)
      IF (AH .GT. -6D0) GOTO 35
      IF ((HH - C8*AH)**2 .GT. R1*R1*ZL) GOTO 35
      II = 1
      GOTO 110
35    T5 = BH - AH
      N1 = 0
      IF (T5 .NE. 0.D0) GOTO 40
      N1 = 1
      GOTO 45
40    J1 = 1
      A = AH
      B = BH
45    CALL TQUA1(AK,BK,N,X,R,L,H,K,S3,S1,S2,PY,T8,DEP,FK)
      IF (AK .GT. -6D0) GOTO 50
      IF ((HK - C8*AK)**2 .GT. R2*R2*ZL) GOTO 50
      II = 1
      J1 = 2
      GOTO 110
50    A2 = BK - AK
      N2 = 0
      IF (A2 .NE. 0.D0) GOTO 55
      N2 = 1
      GOTO 85
55    IF (N1 .NE. 0) GOTO 65
      IF (AH .LT. -2D0 .AND. AK - AH .GT. 4D0) GOTO 85
      IF (AK .LT. -2D0 .AND. AH - AK .GT. 4D0) GOTO 65
      W1 = T5 - A2
      IF (ABS(W1) .GT. 5D2*EPS) GOTO 60
      IF (AH .GT. AK .AND. AH .LT. -2.D0) GOTO 85
60    IF (W1) 65,65,85
65    J1 = 2
      A = AK
      B = BK
      CALL TQUA1(AL,BL,N,X,R,H,K,L,S1,S2,S3,PY,T8,DEP,FL)
      IF (AL .GT. -6D0) GOTO 70
      IF ((HL - C8*AL)**2 .GT. R3*R3*ZL) GOTO 70
      II = 1
      J1 = 3
      GOTO 110
70    T5 = BL - AL
      IF (T5 .NE. 0.D0) GOTO 75
      GOTO 110
75    IF (N2 .NE. 0) GOTO 105
      IF (AK .LT. -2D0 .AND. AL - AK .GT. 4D0) GOTO 110
      IF (AL .LT. -2D0 .AND. AK - AL .GT. 4D0) GOTO 105
      W1 = T5 - A2
```

```
      IF (ABS(W1) .GT. 5D2*EPS) GOTO 80
      IF (AK .GT. AL .AND. AK .LT. -2.D0) GOTO 110
 80   IF (W1) 110,105,105
 85   CALL TQUA1(AL,BL,N,X,R,H,K,L,S1,S2,S3,PY,T8,DEP,FL)
      IF (AL .GT. -6D0) GOTO 90
      IF ((HL - C8*AL)**2 .GT. R3*R3*ZL) GOTO 90
      II = 1
      J1 = 3
      GOTO 110
 90   A2 = BL - AL
      IF (A2 .NE. 0.D0) GOTO 95
      IF (N1 .NE. 1) GOTO 110
      ELLCOV = 0.D0
      RETURN
 95   IF (N1 .EQ. 1) GOTO 105
      IF (AH .LT. -2D0 .AND. AL - AH .GT. 4D0) GOTO 110
      IF (AL .LT. -2D0 .AND. AH - AL .GT. 4D0) GOTO 105
      W1 = T5 - A2
      IF (ABS(W1) .GT. 5D2*EPS) GOTO 100
      IF (AH .GT. AL .AND. AH .LT. -2.D0) GOTO 110
 100  IF (W1) 105,105,110
 105  J1 = 3
      A = AL
      B = BL
C-----------------------------------
 110  CALL CH(J1,H,K,L,S1,S2,S3)
      IF (S1 .NE. S2) GOTO 115
C-----------------------------------
C        S1 = S2,   J = 1.
C-----------------------------------
      J = 1
      D = SQRT(H*H + K*K)/S1
      GO TO 125
 115  IF (H + K .NE. 0.D0) GOTO 125
C-----------------------------------
C        H = K = 0,   J = 0.
C-----------------------------------
      IF (S1 .GT. S2) GO TO 120
      W1 = S1
      S1 = S2
      S2 = W1
 120  D = S2/S1
      J = 0
 125  IF (II .EQ. 0) GOTO 135
C-----------------------------------
C   SEE P.10 OF REPT. 87-27.
C-----------------------------------
      IF (J .LT. 2) GO TO 130
      CALL PKILL(R,S1,S2,H,K,ELLCOV)
```

```
      RETURN
  130 CALL CIRCV(R/S1,D,J,ELLCOV,IER)
      RETURN
C------------------------------------------------
C     PKILL OR CIRCV .GT. 1 - 1E-8 FOR ALL T .GE. T8.
C------------------------------------------------
  135 PY = 0.0D0
      XK7 = DMAX1(S1,S2)
      T8 = C5*XK7 + SQRT(H*H + K*K)
      T8 = R*R - T8*T8
      ZL = L/(C8*S3)
      IF (T8 .LT. 0.0D0) GO TO 140
      A3 = SQRT(T8)
      ZH = A3/(C8*S3)
      T8 = ZL - ZH
      IF (T8 .GE. B) GO TO 140
      B = T8
      PY = AERF(ZL,ZH)/2
      ELLCOV = PY
      IF ((1.D0 - PY) .LT. E1) RETURN
      IF (B - A .LE. ABS(B + A)*DMAX1(1D-9,EPS)) RETURN
C---------------------------------------------
C     OBTAIN SHARP INTEGRATION LIMITS FOR A AND B.
C---------------------------------------------
  140 A2 = B - A
      XK7 = A
      T8 = B
      CALL SQUAD(A,B,W1,N,X,Y,R,H,K,L,S1,S2,S3,EPS,DEP,
     *    J,D,PY,IL)
      IF (W1 .LT. Z4) GOTO 145
C----------------------------------------------------
C     IF A AND B ARE UNCHANGED BY SQUAD, ELLCOV = W1 + PY.
C----------------------------------------------------
      IF(ABS(B - A) .LT. A2*(1.D0 - EPS)) GOTO 150
  145 ELLCOV = PY + W1
      RETURN
C---------------------------------------------
C          START OF GAUSSIAN INTEGRATION.
C---------------------------------------------
C     I1 .GE. 1. GIVES NO. OF SUBINTERVALS OVER WHICH GAUSSIAN
C     NUMERICAL INTEGRATION IS APPLIED. GENERALLY I1 = 1. HIGHER
C     VALUES USED ONLY FOR CHECKING PURPOSES.
C---------------------------------------------
  150 I1 = 1
      CALL RQUAD(A,B,ELLCOV,I1,N,X,Y,R,H,K,L,S1,S2,S3,DEP,J,D)
      ELLCOV = B1*ELLCOV + PY
      IF (ELLCOV .GT. 1.0D0) ELLCOV = 1.0D0
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION BDAW1(XM,XP,DEL,T)
C-----------------------------------------------------------
C     LET F1 = XP/DEL, S0 = XM/DEL.
C     DAW(X) = EXP(-X*X) * INTEGRAL (FROM 0 TO X) EXP(W*W) DW. BDAW1
C     GIVES THE QUANTITY (XM/XP**2) FRAC(XP) - T*(XP/XM**2) FRAC(XM),
C     WHERE FRAC(X) APPEARS IN THE MINIMAX RATIONAL APPROXIMATION
C     GIVEN FOR DAW(X) BY (1/(2X) + 1/(2 X**3)) FRAC(X), WHERE X .GE. 5.
C     BDAW1 IS CALLED FROM SEQHZ3, WHICH IS CALLED BY ELLCOV.
C     EVALUATION OF DAWSON'S INTEGRAL FOR ALL REAL X BY DAW(X)IS BASED
C     ON RATIONAL CHEBYSHEV APPROXIMATIONS PUBLISHED IN MATH. COMP.
C     24, 171-178(1970) BY CODY, PACIOREK AND THACHER.
C-----------------------------------------------------------
      DIMENSION           P4(7)  ,Q4(6)
C-----------------------------------------------------------
      DOUBLE PRECISION    DEL    ,FRM   ,FRP   ,P4    ,Q4    ,T
      DOUBLE PRECISION    XLG    ,XM    ,XP    ,ZM2   ,ZP2
C--------------------------------
      DATA XLG/16777216.0D0/
C-------------------------------------------------------
C         COEFFICIENTS FOR R(6,6) APPROXIMATION,
C         IN J-FRACTION FORM, USED FOR ABS(X) .GT. 5.0
C-------------------------------------------------------
      DATA  P4(1)/-.315576735766984D+02/, P4(2)/-.100791496592972D+02/,
     *      P4(3)/-.710713709224200D+01/, P4(4)/-.596879853243925D+01/,
     *      P4(5)/-.449773645376092D+01/, P4(6)/-.249999965398199D+01/,
     *      P4(7)/ .499999999999330D+00/
      DATA  Q4(1)/ .168874162155616D+03/, Q4(2)/ .698280748271071D+01/,
     *      Q4(3)/-.213029621139181D+02/, Q4(4)/-.712157348463305D+01/,
     *      Q4(5)/-.250005973192356D+01/, Q4(6)/ .750000000715687D+00/
C-------------------------------------------------------
      FRP = 0.0D0
      BDAW1 = 0.0D0
      IF (DEL*XLG .LE. ABS(XM)) RETURN
      IF (DEL*XLG .LE. XP) GOTO 10
      ZP2 = (DEL/XP)**2
      DO 5 I = 1, 6
    5   FRP = ZP2*Q4(I) / (ZP2*(P4(I) + FRP) + 1)
      FRP = P4(7) + FRP
   10 FRM = 0.D0
      IF (T .EQ. 0.D0) GOTO 25
      ZM2 = (DEL/XM)**2
      DO 15 I = 1, 6
   15   FRM = ZM2*Q4(I) / (ZM2*(P4(I) + FRM) + 1)
      FRM = P4(7) + FRM
   20 BDAW1 = XM*FRP/XP**2 - T*XP*FRM/XM**2
      RETURN
   25 BDAW1 = XM*FRP/XP**2
      RETURN
      END
```

```
      SUBROUTINE CH(J,H,K,L,S1,S2,S3)
C-----------------------------------------------------------
C     BASED ON THE VALUE OF J (=1,2,3) H,K,L AND S1,S2,S3 ARE
C     INTERCHANGED. USED IN ELLCOV.
C-----------------------------------------------------------
      DOUBLE PRECISION    K       ,L
C-----------------------------------------------------------
      DOUBLE PRECISION    H       ,S1      ,S2      ,S3      ,X
C-----------------------------------------------------------
      IF (J .EQ. 3) RETURN
      IF (J .EQ. 2) GO TO 5
      X = H
      H = L
      L = X
      X = S1
      S1 = S3
      S3 = X
      RETURN
    5 X = K
      K = L
      L = X
      X = S2
      S2 = S3
      S3 = X
   10 RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION DXDAW(X)
C--------------------------------------------------------------
C   THIS FUNCTION COMPUTES VALUES OF THE FUNCTION -
C      EXP(-X*X) * (1/X)*INTEGRAL (FROM 0 TO X) EXP(T*T) DT,
C   DEFINED FOR ALL REAL ARGUMENTS. USED IN SEQHZ3.

C   THE MAIN COMPUTATION INVOLVES EVALUATION OF RATIONAL CHEBYSHEV
C   APPROXIMATIONS PUBLISHED IN MATH. COMP. 24, 171-178(1970) BY
C   CODY, PACIOREK AND THACHER.
C--------------------------------------------------------------
      DOUBLE PRECISION   P1(9) ,Q1(9) ,P2(8) ,Q2(7) ,P3(8) ,Q3(7) ,P4(7) ,Q4(6)
      DOUBLE PRECISION   FRAC  ,SUMP ,SUMQ  ,W2   ,X    ,Y  ,XLARGE
C--------------------------------------------------------------
      DATA XLARGE/16777216.0D0/
C-----------------------------------------------
C         COEFFICIENTS FOR R(8,8) APPROXIMATION,
C         USED FOR ABS(X) .LT. 2.5
C-----------------------------------------------
      DATA  P1(1)/.100000000000000D+01/, P1(2)/-.135599049815353D+00/,
     *      P1(3)/.456738974064825D-01/, P1(4)/-.258323495918050D-02/,
     *      P1(5)/.360079463580992D-03/, P1(6)/-.944375029163387D-05/,
     *      P1(7)/.634674256878843D-06/, P1(8)/-.711645839183817D-08/,
     *      P1(9)/.977985913592343D-10/
      DATA  Q1(1)/.100000000000000D+01/, Q1(2)/.531067616851310D+00/,
     *      Q1(3)/.133052308640737D+00/, Q1(4)/.206907491644210D-01/,
     *      Q1(5)/.220437428972266D-02/, Q1(6)/.166706801664365D-03/,
     *      Q1(7)/.887964712053131D-05/, Q1(8)/.311750854173480D-06/,
     *      Q1(9)/.574807177698046D-08/
C-----------------------------------------------
C         COEFFICIENTS FOR R(7,7) APPROXIMATION,
C         IN J-FRACTION FORM, USED FOR
C         2.5 .LE. ABS(X) .LT. 3.5
C-----------------------------------------------
      DATA  P2(1)/-.150695651187161D+01/, P2(2)/ .293365747395449D+02/,
     *      P2(3)/-.400000893643550D+02/, P2(4)/-.757931918089369D-01/,
     *      P2(5)/-.889106479747812D+01/, P2(6)/ .152644099623699D+02/,
     *      P2(7)/-.597678086823489D+01/, P2(8)/ .500236896088668D+00/
      DATA  Q2(1)/-.673106069744813D+00/, Q2(2)/ .124486788262252D+04/,
     *      Q2(3)/ .721193217600229D+01/, Q2(4)/ .112461662024575D+03/,
     *      Q2(5)/ .729177556415532D+02/, Q2(6)/ .115840292551888D+03/,
     *      Q2(7)/ .226064666074309D+00/
C-----------------------------------------------
C         COEFFICIENTS FOR R(7,7) APPROXIMATION,
C         IN J-FRACTION FORM, USED FOR
C         3.5 .LE. ABS(X) .LE. 5.0
C-----------------------------------------------
```

```
      DATA  P3(1)/ .476405645273229D+01/, P3(2)/-.266167674896399D+02/,
     *      P3(3)/-.916804879813552D+01/, P3(4)/-.150507703496692D+02/,
     *      P3(5)/ .506460153742231D+01/, P3(6)/-.498544802986608D+01/,
     *      P3(7)/-.149838042036691D+01/, P3(8)/ .499999902705054D+00/
      DATA  Q3(1)/ .287776122973187D+03/, Q3(2)/ .256105722342226D+02/,
     *      Q3(3)/ .751701277744067D+02/, Q3(4)/ .146515167783109D+03/,
     *      Q3(5)/ .330707724676114D+02/, Q3(6)/-.148715811787195D+01/,
     *      Q3(7)/ .250011459611839D+00/
C-----------------------------------------
C       COEFFICIENTS FOR R(6,6) APPROXIMATION,
C       IN J-FRACTION FORM, USED FOR ABS(X) .GT. 5.0
C-----------------------------------------
      DATA  P4(1)/-.315576735766984D+02/, P4(2)/-.100791496592972D+02/,
     *      P4(3)/-.710713709224200D+01/, P4(4)/-.596879853243925D+01/,
     *      P4(5)/-.449773645376092D+01/, P4(6)/-.249999965398199D+01/,
     *      P4(7)/ .499999999999330D+00/
      DATA  Q4(1)/ .168874162155616D+03/, Q4(2)/ .698280748271071D+01/,
     *      Q4(3)/-.213029621139181D+02/, Q4(4)/-.712157348463305D+01/,
     *      Q4(5)/-.250005973192356D+01/, Q4(6)/ .750000000715687D+00/
C-----------------------------------------------------------------
C
      Y = X * X
      IF (ABS(X) .GT. XLARGE) GO TO 35
      IF (Y .GE. 6.25D0) GO TO 5
C-----------------------------------
C         ABS(X) .LT. 2.5
C-----------------------------------
      SUMP = (((((((P1(9) * Y + P1(8)) * Y + P1(7)) * Y + P1(6))
     1      * Y + P1(5)) * Y + P1(4)) * Y + P1(3)) * Y + P1(2))
     2      * Y + P1(1)
      SUMQ = (((((((Q1(9) * Y + Q1(8)) * Y + Q1(7)) * Y + Q1(6))
     1      * Y + Q1(5)) * Y + Q1(4)) * Y + Q1(3)) * Y + Q1(2))
     2      * Y + Q1(1)
      DXDAW = SUMP / SUMQ
      RETURN
C-----------------------------------
C       2.5 .LE. ABS(X) .LT. 3.5
C-----------------------------------
    5 IF (Y .GE. 12.25D0) GO TO 15
      FRAC = 0.0D0
      DO 10 I = 1, 7
   10 FRAC = Q2(I) / (P2(I) + Y + FRAC)
      DXDAW = (P2(8) + FRAC) / Y
      RETURN
C-----------------------------------
C       3.5 .LE. ABS(X) .LT. 5.0
C-----------------------------------
```

```
   15 IF (Y .GE. 25.0D0) GO TO 25
      FRAC = 0.0D0
      DO 20 I = 1, 7
   20 FRAC = Q3(I) / (P3(I) + Y + FRAC)
      DXDAW = (P3(8) + FRAC) / Y
      RETURN
C-----------------------------------
C       5.0 .LE. ABS(X) .LE. XLARGE
C-----------------------------------
   25 W2 = 1.0D0 / X / X
      FRAC = 0.0D0
      DO 30 I = 1, 6
   30 FRAC = Q4(I) / (P4(I) + Y + FRAC)
      FRAC = P4(7) + FRAC
      DXDAW = (0.5D0 + 0.5D0 * W2 * FRAC) / Y
      RETURN
C-----------------------------------
C         XLARGE .LT. ABS(X)
C-----------------------------------
   35 DXDAW = 0.5D0 / Y
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION EQSIG(R3,XK0,XK2,S,EPS,DEP,A5)
C-------------------------------------------------
C     EQSIG COMPUTES ELLCOV FOR EQUAL SIGMAS. XK0 = H*H+K*K+L*L,
C     XK2 = SQRT(XK0). EPS = 10*DPMPAR(1).
C     A = (XK2-R3)*SQRT(.5)/S, A5 = EXP(-A*A).
C-------------------------------------------------
      EXTERNAL  ERF5  ,HSEXP  ,AERF  ,DXPARG
C-------------------------------------------------
      DOUBLE PRECISION   AERF  ,DXPARG
C-------------------------------------------------
      DOUBLE PRECISION   A3    ,A4    ,A5    ,A6    ,C7    ,DEP
      DOUBLE PRECISION   E     ,EPS   ,G1    ,H2    ,H4    ,H5
      DOUBLE PRECISION   H6    ,H7    ,P4    ,P5    ,Q5    ,R3
      DOUBLE PRECISION   S     ,S1    ,T     ,V1    ,XK0   ,XK2
      DOUBLE PRECISION   Z
C-------------------------------------------------
      DATA A4/1.1283 79167 09551 257D0/, C7/.70710 67811 86547 524D0/
C-------------------------------------------------
C         A4 = 2/SQRT(PI)
C-------------------------------------------------
      EQSIG = 0.D0
      H6 = R3*C7/S
      A6 = XK2*C7/S
      A3 = A6 - H6
      E = MAX(1D-11,EPS)
      IF (XK0 .NE. 0.0D0) GOTO 15
      IF (H6 .GT. .071D0) GOTO 10
C-----------------------------------
C     D = 0 AND R3/(SQRT(2)*S) .LE. .071
C-----------------------------------
      N = 3
      EQSIG = 1.D0
      A6 = 1.D0
      P5 = 2.D0*H6*H6
      Q5 = P5*H6/N
    5   N = N + 2
        A6 = A6*P5/N
        EQSIG = EQSIG + A6
        IF (A6 .GT. E*EQSIG) GOTO 5
      EQSIG = A4*A5*Q5*EQSIG
      KKK = 13
      RETURN
C-----------------------------------
C     D = 0 AND R3/(SQRT(2)*S) .GT. .071
C-----------------------------------
   10 CALL ERF5(-A3,A5,P4)
      EQSIG = P4 - A4*H6*A5
      KKK = 14
      RETURN
```

```
C------------------------------------
C    D .NE. 0 AND  R3/(SQRT(2)*S) .GE. .425
C------------------------------------
   15  IF (H6 .LT. .425D0) GOTO 30
       P5 = 4.0D0*A6*H6
       IF (A3 .LT. 4.5D0) GOTO 25
       IF (P5 .LT. DEP)  GOTO 25
       S1 = H6/A6
       E = MAX(2.5D-8,EPS)
       Z = 1.D0
       N = -1
       H7 = 2*A3*A3
   20    N = N + 2
         Z = -N*Z/H7
         S1 = S1 + Z
         IF (ABS(Z) .GT. E*ABS(S1)) GOTO 20
       S1 = S1 - .5D0*Z
       EQSIG = .25D0*A4*A5/A3*S1
       RETURN
   25  P4 = AERF(A6,H6)
       V1 = -1.D0
       CALL HSEXP(-P5,V1,Q5)
       EQSIG = 0.5D0*P4 - A4*H6*A5*Q5
       KKK = 15
       RETURN
C------------------------------------
C    D .NE. 0 AND R3/(SQRT(2)*S) .LT. .425
C------------------------------------
   30  T = A6*A6/2
       KKK = 16
       IF (T + 10 .GT. -DXPARG(1)) RETURN
       G1 = EXP(-T)
       H2 = H6*H6
       H7 = H2*H6
       H5 = 2*G1
       H4 = G1
       N = 1
       V1 = 1.5D0
       S1 = H5*H7/V1
       T = 2*T
       J = 0
   35    H7 = H2*H7
         H4 = (T*H5 - H4)/N
         H5 = (H4 - H5)/V1
         N = N + 1
         V1 = N + .5D0
         Z = H5*H7/V1
         S1 = S1 + Z
```

```
      IF (ABS(Z) .GT. E*ABS(S1)) GOTO 35
        IF (J .GT. 0) GOTO 40
        J = 1
        GOTO 35
40    EQSIG = A4*G1*S1/2
      KKK = 17
      RETURN
      END
```

```
      SUBROUTINE ERF5(X,EXPP,Y)
C-----------------------
C     Y = THE REAL ERROR FUNCTION OF X. EXPP = EXP(-X*X).
C     IF EXP(-X*X) NOT AVAILABLE SET EXPP .LT. 0.
C-----------------------------------------------
      DIMENSION          A(4)   ,B(4)  ,P(8)  ,Q(8)  ,R(5)  ,S(5)
C-----------------------------------------------
      DOUBLE PRECISION   A      ,AX    ,B     ,BOT   ,C     ,EXPP
      DOUBLE PRECISION   P      ,Q     ,R     ,S     ,T     ,TOP
      DOUBLE PRECISION   X      ,X2    ,Y
C-----------------------------------------------
      DATA  C/.564189583547756D0/
      DATA  A(1)/-1.65581836870402D-4/,     A(2)/3.25324098357738D-2/,
     *      A(3)/1.02201136918406D-1/,      A(4)/1.12837916709552D0/
      DATA  B(1)/4.64988945913179D-3/,      B(2)/7.01333417158511D-2/,
     *      B(3)/4.23906732683201D-1/,      B(4)/1.00000000000000D0/
      DATA  P(1)/-1.36864857382717D-7/,     P(2)/5.64195517478974D-1/,
     *      P(3)/7.21175825088309D0/,       P(4)/4.31622272220567D01/,
     *      P(5)/1.52989285046940D02/,      P(6)/3.39320816734344D02/,
     *      P(7)/4.51918953711873D02/,      P(8)/3.00459261020162D02/
      DATA  Q(1)/1.00000000000000D0/,       Q(2)/1.27827273196294D01/,
     *      Q(3)/7.70001529352295D01/,      Q(4)/2.77585444743988D02/,
     *      Q(5)/6.38980264465631D02/,      Q(6)/9.31354094850610D02/,
     *      Q(7)/7.90950925327898D02/,      Q(8)/3.00459260956983D02/
      DATA  R(1)/2.10144126479064D0/,       R(2)/2.62370141675169D01/,
     *      R(3)/2.13688200555087D01/,      R(4)/4.65807828718470D0/,
     *      R(5)/2.82094791773523D-1/
      DATA  S(1)/9.41537750555460D01/,      S(2)/1.87114811799590D02/,
     *      S(3)/9.90191814623914D01/,      S(4)/1.80124575948747D01/,
     *      S(5)/1.00000000000000D0/
C-----------------------------------------------
      AX = ABS(X)
      T = X*X
      IF (AX .GE. 0.5D0) GO TO 5
      TOP = ((A(1)*T + A(2))*T + A(3))*T + A(4)
      BOT = ((B(1)*T + B(2))*T + B(3))*T + B(4)
      Y = X*TOP/BOT
      RETURN
C
    5 IF (AX .GT. 4.0D0) GO TO 10
      TOP = ((((((P(1)*AX + P(2))*AX + P(3))*AX + P(4))*AX + P(5))*AX
     *         + P(6))*AX + P(7))*AX + P(8)
      BOT = ((((((Q(1)*AX + Q(2))*AX + Q(3))*AX + Q(4))*AX + Q(5))*AX
     *         + Q(6))*AX + Q(7))*AX + Q(8)
      IF (EXPP .LT. 0.D0) EXPP = EXP(-T)
      Y = 0.5D0 + (0.5D0 - EXPP*TOP/BOT)
      IF (X .LT. 0.0D0) Y = -Y
      RETURN
```

```
10 Y = 1.0D0
   IF (AX .GE. 5.6D0) GO TO 15
   X2 = 1.0D0/T
   TOP = (((R(1)*X2 + R(2))*X2 + R(3))*X2 + R(4))*X2 + R(5)
   BOT = (((S(1)*X2 + S(2))*X2 + S(3))*X2 + S(4))*X2 + S(5)
   Y = (C - TOP/(T*BOT)) / AX
   IF (EXPP .LT. 0.D0) EXPP = EXP(-T)
   Y = 0.5D0 + (0.5D0 - EXPP*Y)
15 IF (X .LT. 0.0D0) Y = -Y
   RETURN
   END
```

```
      DOUBLE PRECISION FUNCTION FN2(T5,IM,IL,R3,H,K,L,
     *          S1,S2,S3,HZ,DEP,J,D)
C-----------------------------------------------------
C     FN2 GIVES THE INTEGRAND OF ELLCOV. USED BY RQUAD,SQUAD,TQUAD1.
C     J = 0,1,2 SPECIFIES CIRCV OR PKILL. DEP = -DEPSLN(0). D NEEDED
C     FOR CIRCV. HZ = L*SQRT(1/2)/S3. SEE BELOW FOR IM, IL.
C-----------------------------------------------------
C     U(J) ARE USED WHEN SQUAD CALLS FN2  AND THE INPUT INTEGRATION
C     LIMITS A,B ARE UNCHANGED FROM L-R AND L.THEN R, USED IN PKILL
C     OR CIRCV, IS GIVEN BY R3*U(IM).
C     IF IL .NE. 0 COMPUTE FN2 USING THE U(J) USING INPUT IM.
C     IF IL .EQ. 0 COMPUTE FN2 USING T5 WITHOUT USING U(J).
C     U(I)= SQRT(1 - .25*(1 + X(I))**2), I = 12,...,1.
C     U(N+I)= SQRT(1 - .25*(1 - X(I))**2),
C     X(I) ARE THE GAUSSIAN ABSCISSAS OF ORDER 24. N = 12.
C-----------------------------------------------------
      EXTERNAL              CIRCV ,PKILL
      DOUBLE PRECISION      K     ,L     ,U(24)
C-----------------------------------------------------
      DOUBLE PRECISION      C8    ,D     ,DEP   ,H     ,HZ    ,P
      DOUBLE PRECISION      R     ,R3    ,S1    ,S2    ,S3    ,T5
      DOUBLE PRECISION      V2    ,XJ1
C-----------------------------------------------------
       DATAU(12) /.69332454811144340D-1/,  U(11) /.1584669762375325D0/,
     *        U(10) /.2465216831531283D0/ ,  U(9)  /.3322034239275342D0/,
     *        U(8)  /.4146060693752110D0/ ,  U(7)  /.4929421360269080D0/,
     *        U(6)  /.5665216929750006D0/ ,  U(5)  /.6347583153788445D0/,
     *        U(4)  /.6971793487850529D0/ ,  U(3)  /.7534359213923792D0/,
     *        U(2)  /.8033112478280709D0/ ,  U(1)  /.8467264801504051D0/,
     *        U(13) /.8837435290006371D0/ ,  U(14) /.9145642833397272D0/,
     *        U(15) /.9395256076023404D0/ ,  U(16) /.9590894389984678D0/,
     *        U(17) /.9738272897857341D0/ ,  U(18) /.9843985374573839D0/,
     *        U(19) /.9915221334137353D0/ ,  U(20) /.9959418550983111D0/,
     *        U(21) /.9983860184685972D0/ ,  U(22) /.9995236326707759D0/,
     *        U(23) /.9999201660778605D0/ ,  U(24) /.9999971046393888D0/
C-----------------------------------------------------
      DATA C8 /1.4142 13562 37310D0/
C-------------------------------
      FN2 = 0.D0
      IF (T5 .GT. 11.2D0) RETURN
      XJ1 = EXP(-T5*T5)
      P = HZ - T5
      IF (IL .EQ. 0) GOTO 5
      R = R3*U(IM)
      GOTO 10
    5 V2 = C8*S3*T5
      R = (R3 - L)*(R3 + L) + V2*(2*L - V2)
```

```
      IF(R .LT. 0.0D0) RETURN
      R = SQRT(R)
   10 IF (L .NE. 0.0D0) GO TO 15
      XJ1 = 2*XJ1
      GO TO 20
   15 V2 = 4*HZ*P
      IF (V2 .GT. DEP) GO TO 20
      XJ1 = XJ1*((0.5D0 + EXP(-V2)) + 0.5D0)
C-----------------------------------
   20 IF (J .GT. 1) GO TO 25
      CALL CIRCV(R/S1,D,J,P,IER)
      GO TO 30
   25 CALL PKILL(R,S1,S2,H,K,P)
   30 FN2 = XJ1*P
      RETURN
      END
```

```
      SUBROUTINE HSEXP (X ,E, Y)
C-----------------------------------------------------------------------
C              EVALUATION OF (EXP(X) - 1)/X
C
C                 Y = (EXP(X) - 1)/X
C
C  E IS AN INPUT/OUTPUT VARIABLE. IF E IS .GE. 0 THEN IT IS ASSUMED
C  THAT E = EXP(X). IN THIS CASE E IS NOT MODIFIED. IF E IS NEGATIVE
C  THEN E IS SET TO EXP(X) WHEN THIS VALUE IS NEEDED IN HSEXP.
C-----------------------------------------------------------------------
      DOUBLE PRECISION   X      ,P1     ,P2     ,Q1     ,Q2     ,Q3
      DOUBLE PRECISION   Q4     ,E      ,Y
C-----------------------------------------------------------------------
      DATA  P1/ .914041914819518D-09/,      P2/ .238082361044469D-01/,
     *      Q1/-.499999999085958D+00/,      Q2/ .107141568980644D+00/,
     *      Q3/-.119041179760821D-01/,      Q4/ .595130811860248D-03/
C-----------------------------------------------------------------------
      IF (ABS(X) .GT. 0.15D0) GO TO 5
      Y = ((P2*X + P1)*X + 1.0D0)/((((Q4*X + Q3)*X + Q2)*X
     *            + Q1)*X + 1.0D0)
      RETURN
C
    5 IF (E .LE. 0.D0) E = EXP(X)
      IF (X .GT. 0.0D0) GO TO 10
        Y = ((E - 0.5D0) - 0.5D0)/X
        RETURN
   10 Y = E*(0.5D0 + (0.5D0 - 1.0D0/E))/X
      RETURN
      END
```

```
      SUBROUTINE RQUAD(A,B,XI1,M,N,X,Y,R3,H,K,L,S1,S2,S3,DEP,J,D)
C----------------------------------------------------------
C     RQUAD IS A QUADRATURE ROUTINE WHICH USES GAUSSIAN MULTIPLIERS
C     TO OBTAIN THE INTEGRAL C FROM A TO B OF FN2(T). A, B ARE THE
C     LOWER AND UPPER LIMITS OF INTEGRATION. OUTPUT IS XI1 = C.
C     INPUT IS M, NO. OF EQUAL SUBDIVISIONS OF [A,B] WITH SAME ORDER
C     GAUSSIAN INTEGRATION APPLIED ON EACH SUBDIVISION. M SET TO 1;
C     HIGHER VALUES OF M USED ONLY FOR CHECKING PURPOSES.
C     N, (2N = 0RDER OF GAUSSIAN MULTIPLIERS USED). SET AT N = 12.
C     X(*),Y(*)--STORED VALUES OF GAUSSIAN ABSCISSAS AND MULTIPLIERS
C     ON (-1,1}. REMAINING ARGUMENTS OF THE CALL LINE ARE USED AS
C     INPUT PARAMETERS FOR FN2, (DEP = -DEPSLN(0)).
C     J = 0,1,2 FOR CIRCV OR PKILL. D NEEDED FOR CIRCV.
C----------------------------------------------------------
      DIMENSION         X(1)   ,Y(1)
      EXTERNAL          FN2
C----------------------------------------------------------
      DOUBLE PRECISION  FN2    ,K     ,L
      DOUBLE PRECISION  A      ,B     ,C7     ,D     ,DEP    ,D3
      DOUBLE PRECISION  D4     ,E2    ,F      ,G     ,H      ,HZ
      DOUBLE PRECISION  R3     ,S1    ,S2     ,S3    ,T      ,TM
      DOUBLE PRECISION  TM1    ,TP    ,TP1    ,X     ,XI1    ,Y
C----------------------------------------------------------
      DATA C7/.7071067811865475D0/
C----------------------------------------
      HZ = C7*L/S3
    5 G = 0.D0
      K1 = 0
      D3 = B - A
      D4 = D3/M
      D3 = D4/2
      E2 = A + D3
   10 K1 = K1 + 1
      HZ = C7*L/S3
C----------------------------------
C     START GAUSSIAN INTEGRATION.
C----------------------------------
      I = N + 1
   15    I = I - 1
         T = D3*X(I)
         TM1 = -T + E2
         TM = FN2(TM1,I,0,R3,H,K,L,S1,S2,S3,HZ,DEP,J,D)
         TP1 = T + E2
         TP = FN2(TP1,I,0,R3,H,K,L,S1,S2,S3,HZ,DEP,J,D)
         F = Y(I)*(TM + TP)
         G = G + F
```

```
   IF (I .GT. 1) GOTO 15
   E2 = E2 + D4
IF (K1 .NE. M) GOTO 10
XI1 = D3*G
RETURN
END
```

```
      SUBROUTINE SEQHZ3(R,L,S1,S3,EPS,DEP,XK0,AA,EQS,SEQ,KKK)
C-----------------------------------------------------------
C     SEQ GIVES THE TRIVARIATE NORMAL PROBABILITY (ELLCOV) OVER
C     A SPHERE WITH CENTER (0,0,L) AND RADIUS R. THE NORMAL DIST-
C     TRIBUTION HAS CORRESPONDING STANDARD DEVIATIONS S1 = S2, S3.
C     XK0 = (L)**2. AA = EXP(-((L-R)/(SQRT(2)*MAX(S1,S3))**2)).
C     EQS = P FOR EQUAL SIGMAS (S1 = S3). KKK INDENTIFIES PATHS,
C     USED FOR EASE IN FOLLOWING PROGRAM. EPS = 10*DPMPAR(1).
C-----------------------------------------------------------
C                 S1 = S2 AND H = K = 0.
C-----------------------------------------------------------
      EXTERNAL            AERF   ,BDAW1 ,DAW   ,DXDAW ,DXPARG
      EXTERNAL            ERF    ,ERFCR ,ERFC0 ,HSEXP ,EQSIG
C-----------------------------------------------------------
      DOUBLE PRECISION    AERF     ,BDAW1  ,DAW    ,DXDAW
      DOUBLE PRECISION    DXPARG   ,ERF    ,ERFCR  ,EQSIG
C-----------------------------------------------------------
      DOUBLE PRECISION    L
C-----------------------------------------------------------
      DOUBLE PRECISION    AA    ,B1    ,C5    ,C7    ,DEP   ,E
      DOUBLE PRECISION    EPS   ,EQS   ,ERL   ,ET    ,E1    ,F1
      DOUBLE PRECISION    R     ,S     ,SEQ   ,S0    ,S1    ,S3
      DOUBLE PRECISION    T     ,T0    ,T1    ,T3    ,U     ,U0
      DOUBLE PRECISION    U1    ,U2    ,V1    ,V2    ,V3    ,V4
      DOUBLE PRECISION    W9    ,X     ,XK0   ,XL1   ,XL2   ,X2
      DOUBLE PRECISION    Y     ,Y1    ,Y2    ,Z     ,ZL    ,ZL1
      DOUBLE PRECISION    ZM    ,ZMN   ,ZP    ,ZPN   ,ZR    ,ZR1
      DOUBLE PRECISION    Z1
C-------------------------------------------------------------
       DATA B1 /.56418 95835 47756D0/, C7/.70710 67811 86548D0/
       DATA C5 /11.2D0/
C---------------------
       E = MAX (EPS/2,1D-10)
       E1 = 2D1*EPS
       SEQ = 0.D0
       ZL = L*C7/S3
       ZR = R*C7/S3
       Y1 = 4*ZL*ZR
       X2 = ZR*ZR
       V1 = R*R/(S1*S1)
C---------------------------------------------------------
C     ZL = L/(SQRT(2)*S3)  ZR = R/(SQRT(2)*S3)  V1 = R*R/(S1*S1)
C---------------------------------------------------------
       IF (Y1 .GT. 1.D1) GOTO 15
       IF (X2 .GT. 2.D0) GOTO 15
       IF (V1 .GT. 2.D0) GOTO 15
```

```
C--------------------------------------------------
C    4*ZL*ZR .LE. 10,     ZR*ZR .LE. 2,      V1 .LE. 2.
C--------------------------------------------------
      F1 = EXP(-ZL*ZL/2)
      S = .66666 66666 666667D0*F1
      T = S
      Y = 2*F1
      Z = .5D0*Y1
      Y1 = Y1*F1
      Z1 = 2*X2
      J = 0
      K = 1
   5     K = K + 1
         Y = (Z*Y1 - Z1*Y)/K
         K = K + 1
         Y1 = (Z*Y - Z1*Y1)/K
         T = (Y/K - V1*T)/(K + 2)
         S = S + T
         IF (ABS(T) .GT. E*ABS(S)) GOTO 5
         IF (J .NE. 0) GOTO 10
         J = 1
      GOTO 5
  10  SEQ = B1*ZR*V1*S*F1
      KKK = 1
      RETURN
C------------------------------------------------------------
C     EITHER   4*ZL*ZR .GT. 10 .OR. ZR*ZR .GT. 2  .OR.  V1 .GT. 2.
C------------------------------------------------------------
  15  ZM = C7*(L - R)/S3
      KKK = 0
      IF (ZM .GT. C5) RETURN
      V2 = S3*S3/(S1*S1)
      XL1 = ABS(.5D0 + (.5D0 - V2))
      IF (S3 .LT. S1) AA = EXP(-ZM*ZM)
      IF (XL1 .GT. E) GOTO 20
      SEQ = EQS
C-----------------------------------
C           XK0 = L*L
C-----------------------------------
      KKK = 4
      IF (R .GT. L) RETURN
      SEQ = EQSIG(R,XK0,L,S3,E1,DEP,AA)
      RETURN
  20  XL2 = SQRT(XL1)
      JJJ = -1
      IF (S1 .LT. S3) JJJ = 1
  25  Z = ZR*ZR*XL1
      IF (ZL .NE. 0.D0 .OR. Z .GT. 3D0 ) GOTO 40
```

```
C------------------------------------------------
C  L .EQ. 0  AND   R*C7/S3)**2*(ABS(1-(S3/S1)**2)).LE. 3
C------------------------------------------------
      K = 3
      JJ = -2*JJJ
      S = JJ*Z/3
      T = S
   30 K = K + 2
      T = JJ*Z*T/K
      S = S + T
      IF (ABS(T) .GT. E*ABS(S)) GOTO 30
      IF (S3 .GT. S1) GOTO 35
      EQS = EQSIG(R,XK0,L,S3,E1,DEP,AA)
   35 SEQ = EQS - 2*ZR*B1*AA*S
      KKK = 5
      RETURN
C------------------------------------------------
C   L .NE. 0  OR    R*C7/S3)**2*(ABS(1-(S3/S1)**2)).GT. 3
C------------------------------------------------
   40 Y = C7*(L - R + V2*R)/S3
      Y2 = C7*(L + R - V2*R)/S3
      IF (V2 .LE. 1.D0) GOTO 45
      T = Y
      Y = Y2
      Y2 = T
   45 S0 = Y/XL2
      F1 = Y2/XL2
      IF (V2 .GT. 1D-2 .OR. V2*MAX(ZL,ZR) .GT. .5D0) GOTO 65
C------------------------------------------------------
C   S3/S1 .LE. .1D0 .AND. MAX(ZL,ZR)*(S3/S1)**2 .LE. 1/2.
C------------------------------------------------------
      X = (R - L)*(R + L)/(2*S1*S1)
      T0 = 0.D0
      IF (ZM .GT. 10.5D0) GOTO 60
      T0 = AERF(ZL,ZR)
      ET = EXP(-X)
      IF (ET .EQ. 0.D0) GOTO 60
C------------------------------
C      HSEXP COMPUTES (EXP(X)-1)/X
C------------------------------
      CALL HSEXP(-X,ET,SEQ)
      SEQ = X*SEQ*T0
C------------------------------
C      AA = EXP(-((L-R)*C7/S3)**2)
C------------------------------
      ERL = 0.D0
      ZPN = 0.D0
```

```
      T1 = B1*AA
      T3 = T1
      IF (Y1 .GT. -DXPARG(1)) GOTO 50
      ERL = EXP(-Y1)
      CALL HSEXP(-Y1,ERL,T1)
      T1 = T3*Y1*T1
      ZPN = 1.D0
   50 U0 = 1.D0
      V3 = 2*V2
      U1 = -V3*ZL
      S = T1*U1
      ZP = ZL + ZR
      ZPN = 1.D0
      ZMN = 1.D0
      N = 1
C--------------------------------
   55    N = N + 1
         IF (N .GT. 20) GOTO 65
         ZPN = ZPN*ZP/N
         ZMN = ZMN*ZM/N
         T0 = .5D0*T0/N + T3*(ZMN - ERL*ZPN)
         U2 = V3*(-ZL*U1 + (N - 1)*U0)
         V4 = T0*U2
         U0 = U1
         U1 = U2
         N = N + 1
         ZPN = ZPN*ZP/N
         ZMN = ZMN*ZM/N
         T1 = .5D0*T1/N + T3*(ZMN - ERL*ZPN)
         U2 = V3*(-ZL*U1 + (N - 1)*U0)
         V4 = V4 + U2*T1
         S = S + V4
         U0 = U1
         U1 = U2
         IF (ABS(V4) .GT. ABS(SEQ - ET*S)*E) GOTO 55
      SEQ = (SEQ - ET*S)/2
      KKK = 2
      RETURN
   60 SEQ = T0/2
      KKK = 3
      RETURN
C--------------------------------------------------
C V2=(S3/S1)**2 .GT. 1D-2 .OR. V2*MAX(ZL,ZR) .GT. .5D0)
C--------------------------------------------------
   65 IF (S0 .LT. 5D0) GOTO 85
      Z1 = B1*ZR*XL1
      W9 = EXP(-Y1)
```

```
      CALL HSEXP(-Y1,W9,T)
      Z = 4*Z*T - JJJ*(1 + W9)
      U = AA/(2*Y*Y2)
      IF (R .GT. L .AND. S3 .GT. S1) GOTO 70
      EQS = EQSIG(R,XK0,L,S3,E1,DEP,AA)
   70 IF (JJJ .GT. 0) GOTO 80
      ZL1 = 0.D0
      IF (W9 .EQ. 0.D0) GOTO 75
      ZL1 = Y*ERFCR(F1)
   75 SEQ = EQS - U*(Z1*Z - Y2*ERFCR(S0) + W9*ZL1)
      KKK = 6
      RETURN
   80 SEQ = EQS - B1*U*XL1*(ZR*Z + BDAW1(Y,Y2,XL2,W9))
      KKK = 7
      RETURN
C-----------------------------------------------------
C     USE (13) - (15) OF REPORT 87-27
C-----------------------------------------------------
   85 SEQ = AERF(ZL,ZR)
      ZL1 = ZL/XL2
      ZR1 = XL2*ZR
      J = -5
      IF (JJJ .GT. 0) GOTO 110
      IF (L .EQ. 0D0) GOTO 105
   90 IF (S0 .LE. 0.D0) GOTO 100
      W9 = EXP(-Y1)
      Y = -1.D0
      Y2 = 0.D0
      CALL ERFC0(1,S0,Y,Y1)
      IF (W9 .EQ. 0.D0) GOTO 95
      CALL ERFC0(1,F1,Y,Y2)
   95 SEQ = .5D0*(SEQ - AA/XL2*(Y1 - W9*Y2))
      KKK = 8
      RETURN
  100 X2 = AERF(ZL1,ZR1)
      SEQ = .5D0*(SEQ - X2*EXP(-.5D0*(R*R - L*L/XL1)/(S1*S1))/XL2)
      KKK = 9
      RETURN
  105 SEQ = SEQ/2 - EXP(-V1/2)*ERF(ZR*XL2)/XL2
      KKK = 10
      RETURN
C-----------------------------------------------------
C    S1 .LT. S3---USE (16) OF REPORT 87-27
C-----------------------------------------------------
  110 J = -6
      IF (L .EQ. 0D0) GOTO 120
      Y = 0.D0
```

```
      W9 = EXP(-Y1)
      IF (W9 .EQ. 0.D0) GOTO 115
      Y = DAW(S0)
115   SEQ = 0.5D0*SEQ - B1*AA/XL2*(DAW(F1) - W9*Y)
      KKK = 11
      RETURN
120   SEQ = .5D0*SEQ- 2*B1*AA*ZR*DXDAW(ZR*XL2)
      KKK = 12
      RETURN
      END
```

```
      SUBROUTINE SQUAD(A,B,Z1,N,X,Y,R3,H,K,L,S1,S2,S3,EPS,DEP,
     *    J,D,PY,IL)
C------------------------------------------------------------
C     SQUAD IS A QUADRATURE ROUTINE WHICH USES GAUSSIAN MULTIPLIERS
C     OF ORDER (24) FOR AN ESTIMATE OF THE INTEGRAL Z1 FROM A TO B
C     OF FN2(T). A, B ARE THE LOWER AND UPPER LIMITS OF INTEGRATION.
C     THE OUTPUT ARE A, B, Z1, WHERE A,B MAY HAVE NEW VALUES, SUCH
C     THAT FN2 IS NEGLIGIBLE ON (A,A(NEW)), AND (B(NEW),B), THAT IS
C     WHEN FN2(T) .LT. 1E-8*MIN(MAX(G, PY, FA, FB), 1/2).
C     EPS = 10*DPMPAR(1).  DEP = - DEPSLN(0). IL IS INFO PARAMETER.
C     J = 0,1,2 SPECIFIES CIRCV OR PKILL. CIRCV NEEDS D IN FN2.
C------------------------------------------------------------
       DIMENSION          X(1)   ,Y(1)   ,TM(24)
       EXTERNAL           FN2
      DOUBLE PRECISION    DEP    ,FN2    ,K      ,L
C------------------------------------------------------------
      DOUBLE PRECISION    A      ,B      ,B1     ,C7     ,D      ,D1
      DOUBLE PRECISION    EPS    ,E3     ,FA     ,FB     ,G      ,H
      DOUBLE PRECISION    HZ     ,PY     ,RZ     ,R3     ,S1     ,S2
      DOUBLE PRECISION    S3     ,TM     ,TM1    ,X      ,Y      ,ZT
      DOUBLE PRECISION    Z1     ,Z2
C------------------------------------------------------------
        DATA B1 /.5641895835477563D0/, C7/.7071067811865475D0/
C------------------------------------------------------------
       IL = 0
       ZT = 0
       FA = 0.D0
       G = 0.D0
       NT = 2*N
       RZ = C7*R3/S3
       HZ = C7*L/S3
       IF (ABS(HZ - B) .GT. EPS*HZ) GOTO 5
       Z2 = HZ - RZ
       IF (ABS(Z2 - A) .LE. EPS*ABS(Z2)) IL = 1
    5  E3 = (B - A)/2
       D1 = (B + A)/2
       J1 = 0
       I = N + 1
   10    I = I - 1
         J1 = J1 + 1
         Z2 = E3*X(I)
         TM1 = -Z2 + D1
         TM(J1) = Y(I)*FN2(TM1,I,IL,R3,H,K,L,S1,S2,S3,HZ,DEP,J,D)
         TM1 = Z2 + D1
         M = I + N
```

```
        TM(M) = Y(I)*FN2(TM1,M,IL,R3,H,K,L,S1,S2,S3,HZ,DEP,J,D)
        G = G + (TM(J1) + TM(M))
        IF (I .GT. 1) GOTO 10
      IF (IL .EQ. 0) FA = FN2(A,I,0,R3,H,K,L,S1,S2,S3,HZ,DEP,J,D)
      FB = FN2(B,M,0,R3,H,K,L,S1,S2,S3,HZ,DEP,J,D)
      Z1 = E3*B1*G
      IF (Z1 + PY .LT. 1D-37) RETURN
      ZT = 1D-8*MIN(MAX(G,PY,FA,FB),.5D0)
      IF (FA .GT. ZT) GOTO 25
C-------------------------------
C     ATTEMPT TO FIND LARGER A
C-------------------------------
      II = 0
   15    II = II + 1
         IF (TM(II) .GT. ZT) GOTO 20
         IF (II .LT. NT) GOTO 15
      Z1 = 0D0
      RETURN
   20 IF (II .EQ. 1) GOTO 25
      I = II - N -2
      IF (I .GE. 0) I = I + 1
      JJ = IABS(I)
      A = ISIGN(1,I)*E3*X(JJ) + D1
   25 IF (FB .GT. ZT) RETURN
C-------------------------------
C     ATTEMPT TO FIND SMALLER B
C-------------------------------
      II = NT + 1
   30    II = II - 1
         IF (TM(II) .GT. ZT) GOTO 35
         IF (II .GT. 1) GOTO 30
   35 IF (II .EQ. NT) RETURN
      I = II - N + 1
      IF (I .LE. 0) I = I - 1
      JJ = IABS(I)
      B = ISIGN(1,I)*E3*X(JJ) + D1
      RETURN
      END
```

```
      SUBROUTINE TQUA1(A,B,N,X,R3,H,K,L,S1,S2,S3,PY,T8,DEP,F)
C-----------------------------------------------------------
C     TQUA1 IS A ROUTINE THAT FINDS THE SMALLEST VALUE OF T, SAY,C ON
C     (A,T1,...,Tk,...,B) FOR WHICH FN2(T) .GE. T8, AND LARGEST VALUE
C     SAY C1, FOR WHICH STARTING FROM B IN DECREASING T, FN2(C1) .GE.
C     T8, WHERE T8 = MAX(1D2*DPMPAR(2),1D-42).
C     TI = ISIGN(1,I)*E3*X(ABS(I)) + D3,
C     WITH X(I) THE GAUSSIAN ABSCISSAS OF ORDER 2*N, E3 = (B - A)/2,
C     AND D3 = (B + A)/2, X's = GAUSSIAN ABSCISSAS OF O(24) USED.
C     IF C=T(L+1) AND C1=T(K-1), THE OUTPUT A=T(L) AND B=T(K) AND
C     F = FN2(T(L)). DEP = -DEPSLN(0).
C-----------------------------------------------------------
      DIMENSION           X(1)
      EXTERNAL            FN2
      DOUBLE PRECISION    FN2   ,K    ,L
C-----------------------------------------------------------
      DOUBLE PRECISION    A     ,B    ,C7   ,D    ,DEP  ,D3
      DOUBLE PRECISION    E3    ,F    ,H    ,HZ   ,PY   ,R3
      DOUBLE PRECISION    S1    ,S11  ,S2   ,S22  ,S3   ,TM
      DOUBLE PRECISION    TM1   ,T8   ,T9   ,X
C-----------------------------------------------------------
       DATA C7/.7071067811865475D0/
C------------------------------------------
       D = 0.D0
       HZ = C7*L/S3
       S11 = S1
       S22 = S2
       J = 2
       TM1 = A
       TM = 0.D0
       IF (S1 .NE. S2) GOTO 5
       J = 1
       D = SQRT(H*H + K*K)/S1
       GOTO 15
    5  IF (H + K .NE. 0.D0) GOTO 15
       J = 0
       IF (S1 .GT. S2) GOTO 10
       D = S11
       S11 = S22
       S22 = D
   10  D = S22/S11
   15  NT = 2*N + 1
       E3 = (B - A)/2
       D3 = (B + A)/2
       T9 = T8
       IF (PY .GT. 0D0) T9 = MIN(T8,PY)
```

```
      DO 20 II = 1, NT
         I = II - 13
         IF (I .EQ. 0) GOTO 20
         A = TM1
         F = TM
         JJ=IABS(I)
         TM1 = ISIGN(1,I)*E3*X(JJ) + D3
         TM = FN2(TM1,I,0,R3,H,K,L,S11,S22,S3,HZ,DEP,J,D)
         IF (TM .GT. T9) GOTO 25
20    CONTINUE
25      TM1 = B
      DO 30 II = 1, NT
         I = II - 13
         IF (I .EQ. 0) GOTO 30
         B = TM1
         JJ = IABS(I)
         TM1 = -ISIGN(1,I)*E3*X(JJ) + D3
         IF (TM1 .LE. A) GOTO 35
         TM = FN2(TM1,I,0,R3,H,K,L,S11,S22,S3,HZ,DEP,J,D)
         IF (TM .GT. T9) GOTO 35
30    CONTINUE
35    IF (TM1 .LE. A) B = A
      RETURN
      END
```

APPENDIX C

FORTRAN LISTINGS FOR ELINV3 AND SUPPORTING ROUTINES

FORTRAN LISTINGS FOR ELINV3 AND SUPPORTING ROUTINES

Below we give a summary of the various subprograms that are used for computing $\widetilde{R}$. These subprograms are listed in this appendix, except for those which are contained in NSWCLIB [13]. The NSWCLIB routines are identified below with a superscript $*$. All subprograms are given in double precision.

REFERENCING OF ROUTINES USED TO COMPUTE $\widetilde{R}$

ELINV3	uses:	DPMPAR*	ELLCOV	ELLRC	GAMINV*	SUB3
ELLRC	uses:	AERF*	DEPSLN*	DPMPAR*	FCN1	
FCN1	uses:	AERF*				
SUB3	uses:	ELLCOV				

```
      SUBROUTINE ELINV3(P3,HX,HY,HZ,SX,SY,SZ,R,PXD,IND,NN)
C-----------------------------------------------------------------
C     (X,Y,Z) IS A POINT IN A CARTESIAN COORDINATE SYSTEM. ELINV3
C     RETURNS R, THE RADIUS OF THE SPHERE WITH CENTER (HX,HY,HZ) WHICH
C     HOLDS P3 OF THE NORMAL ELLIPSOIDAL DISTRIBUTION WITH MEAN (0,0,0)
C     AND STANDARD DEVIATIONS SX, SY, SZ IN X,Y,Z DIRECTIONS, RESPECT-
C     IVELY. ESTIMATES OF P3, FOR A GIVEN R, P(R), ARE OBTAINED FROM
C     ELLCOV. R IS GENERALLY CORRECT TO AT LEAST 6 SIGNIFICANT DIGITS.
C     LET E2 = 10*DPMPAR(1). THE INPUT P3 SHOULD SATISFY 1D-20 .LE. P3
C     .LE. MIN(1 - E2, .99999999). IF P3 IS IN (0,1), AND THE ABOVE
C     INEQUALIITIES ARE NOT SATISFIED, OUTPUT R MAY NOT BE CORRECT TO 6
C     SIGNIFCANT DIGITS. IND = -1, IF P3 .LT. MAX(1D-40, 1E6*DPMPAR(2)),
C     P3 .NE. 0. IND = 1, IF P3 .GT. 1 - MAX(1D-12, E2); R SET TO -1D10, IF
C     ABS(IND) = 1. NN = THE NO. OF ITERATIONS USED; NN .LE. 30, NN
C     AVERAGES 6. IND = 2, IF NN = 30. LET RH AND RL DENOTE THE
C     CURRENT UPPER AND LOWER BOUNDS FOR R. IND = 3 IF (RH - RL) .LE.
C     MAX(E2, 1D-14)*R AND ABS(P(R) - P3) .GT. MAX(E2, 1E-8)*P3; THE
C     PRECISION OF THE MACHINE IS NOT ADEQUATE TO REVERSE THE LAST
C     INEQUALITY. IF IND = 4, THE VALUE OF R IS ACCEPTABLE BUT ELLCOV
C     CANNOT BE COMPUTED WITH SUFFICIENT ACCURACY TO DETERMINE R WITH
C     FULL ACCURACY.
C     REF: INTEGRATION OF THE TRIVARIATE NORMAL DISTRIBUTION OVER AN
C     OFFSET SPHERE AND AN INVERSE PROBLEM. NSWC TR. 87-27, 2/1988.
C     REF: SIGNIFICANT DIGIT COMPUTATION OF THE ELLIPSOIDAL COVERAGE
C     FUNCTION AND ITS INVERSE. NAVSWC TR 91-487.
C-----------------------------------------------------------------
      DIMENSION          A6(17) ,B6(17)
      EXTERNAL           DPMPAR ,ELLCOV ,ELLRC ,GAMINV ,SUB3
C-----------------------------------------------------------------
      DOUBLE PRECISION   DPMPAR ,ELLCOV ,ELLRC
C-----------------------------------------------------------------
      DOUBLE PRECISION   A6     ,B6     ,C1     ,C3     ,C4     ,D3
      DOUBLE PRECISION   D4     ,D5     ,E1     ,E2     ,E3     ,E4
      DOUBLE PRECISION   E5     ,E6     ,E8     ,F0     ,F1     ,GX
      DOUBLE PRECISION   GY     ,GZ     ,HX     ,HY     ,HZ     ,PH
      DOUBLE PRECISION   PHD    ,PL     ,PLD    ,PX     ,PXD    ,PXDO
      DOUBLE PRECISION   P3     ,R      ,RH     ,RL     ,RP     ,R1
      DOUBLE PRECISION   R2     ,S      ,SX     ,SY     ,SZ     ,V4
      DOUBLE PRECISION   V5     ,WX     ,WY     ,WZ     ,W2     ,W4
      DOUBLE PRECISION   XK0    ,XK2    ,XM0    ,X0     ,X1     ,XGAMIN
C-----------------------------------------------------------------
      DATA A6(1)/1D-30/A6(2)/1D-25/A6(3)/1D-20/A6(4)/1D-15/A6(5)/1D-10/
     +     A6(6)/1D-8/A6(7)/5D-6/A6(8)/1D-4/A6(9)/1D-2/A6(10)/1D-1/
     +     A6(11)/3D-1/A6(12)/0.6D0/A6(13)/0.9D0/A6(14)/0.999D0/
     +     A6(15)/.999999D0/A6(16)/.99999999D0/A6(17)/1.0D0/
C-----------------------------------------------------------------
```

```
      DATA B6(1)/1.56D-10/B6(2)/7.23D-9/B6(3)/3.36D-7/B6(4)/1.56D-5/
     +     B6(5)/7.22D-4/B6(6)/3.36D-3/B6(7)/2.66D-2/B6(8)/7.23D-2/
     +     B6(9)/3.39D-1/B6(10)/.765D0/B6(11)/1.1933D0/B6(12)/1.717D0/
     +     B6(13)/2.5005D0/B6(14)/4.0335D0/B6(15)/5.538D0/B6(16)/
     +     6.35D0/B6(17)/7.7D0/
C-----------------------------------------------------------------
      DATA C4/1D-20/C3/0.797884560802865D0/C1/.5773502691896258D0/
C------------------------------------
C     C3 = SQRT(2/PI),  C1 = 1/SQRT(3)
C------------------------------------
CT     R,RL,RH,NN,PX,X0,X1,F1,PXD,F0
      IND = 0
      R = 0D0
      IF (P3 .EQ. 0.0D0) RETURN
      PXDO = -10D0
      XK0 = HX*HX + HY*HY + HZ*HZ
      XK2 = SQRT(XK0)
      S = MAX(SX,SY,SZ)
      E1 = 1D6*DPMPAR(2)
      E2 = 10*DPMPAR(1)
      E3 = MAX(E2,5D-9)
      E4 = MAX(E2,1D-14)
      E5 = MAX(E1,C4*C4)
      E6 = MAX(E2,1D-8)*P3
      W4 = SX*SX + SY*SY + SZ*SZ
      R = -1D10
      IF (P3 .LT. E5) GOTO 175
      IF (P3 .GT. 1.0D0 - MAX(1D-12,E2)) GOTO 180
C-----------------------------------------
      NN = 0
      E8 = E6
      IF (P3 .LT. 0.999D0) GOTO 5
      E8 = MAX(E2, 1.0D-9)
      IF (P3 .LT. 0.999999D0) GOTO 5
      E8 = MAX(E2, 1.D-10)
      IF (P3 .LT. 0.99999999D0) GOTO 5
      E8 = MAX(E2,1D-11)
C-----------------------------
C     FIRST ESTIMATE FOR RMIN
C-----------------------------
    5 PL = P3*S/C3
      RL = MAX(3.0D0*P3*SX*SY*SZ/C3, PL*PL*PL)
      IF (P3 .GE. 0.5D0) GOTO 10
      R = RL**(1.D0/3.D0)
      GOTO 15
   10 R = XK2
      IF (RL .GT. XK0*XK2) R=RL**(1.D0/3.D0)
```

```
 15  RL = R
     PL = 0D0
     PLD = - P3
C----------------------------
C  FIRST ESTIMATE FOR RMAX
C----------------------------
     DO 20 J = 1, 17
        IF (P3 .LE. A6(J)) GOTO 25
 20  CONTINUE
 25  R = XK2 + B6(J)*S
     RH = R
     PH = 1D0
     PHD = 1D0 - P3
C--------------------------
C  GRUBB'S ESTIMATE FOR R
C--------------------------
     KG = 0
     XM0 = 1.0D0 + XK0/W4
     WX = SX*SX/W4
     WY = SY*SY/W4
     WZ = SZ*SZ/W4
     GX = HX/SX
     GY = HY/SY
     GZ = HZ/SZ
     V4 = 2.0D0*(WX*WX*(1.0D0 + 2.0D0*GX*GX) + WY*WY*(1.0D0 + 2.0D0*
    +      GY*GY) + WZ*WZ*(1.0D0 + 2.0D0*GZ*GZ))
     V5 = 8.0D0*(WX*WX*WX*(1.0D0 + 3.0D0*GX*GX) + WY*WY*WY
    +       *(1.0D0 + 3.0D0*GY*GY) + WZ*WZ*WZ*(1.0D0 + 3.0D0*GZ*GZ))
     W2 = 0.5D0*V5/V4
     V5 = V5*V5/(V4*V4*V4)
C----------------------------
     I2 = 0
     D3 = 1.1D0
     IF (P3 .LE. 0.8D0) GOTO 30
     D3 = 1.25D0
     IF (P3 .LE. 0.9D0) GOTO 30
     D3 = 1.9D0
 30  D4 = P3
     PX = P3
C------------------------------------------
 35  RP = R
     IF (PX .EQ. 1.D0) PX = 1.D0 - E8
     CALL GAMINV(4.D0/V5, XGAMIN, 0.D0, PX, 1.D0 - PX, IERR)
     R = W4*((XGAMIN - 4.0D0/V5)*W2 + XM0)
     IF (R .LE. RL*RL .OR. R .GE. RH*RH) GOTO 75
     R = SQRT(R)
     IF (KG .EQ. 0) GOTO 40
     IF (ABS(R - RP) .LT. E3*R) GOTO 75
```

```
 40  CALL SUB3(P3,R,HX,HY,HZ,SX,SY,SZ,XK0,XK2,S
   +          ,PX,PXD,RL,PL,PLD,RH,PH,PHD,NN,KK)
     KG = 1
 45  IF (ABS(PXD) .LE. E8) RETURN
     IF (PXD .LT. 0.D0) GOTO 60
C-------------------------
C       PX .GE. P3
C-------------------------
     IF (I2 .LT. 0) GOTO 100
     IF (NN .GT. 4) GOTO 75
     IF (PX .GT. 10.0D0*P3) GOTO 75
     IF (P3 .GT. 0.01D0 .OR. PX .EQ. 1.0D0) GOTO 50
     IF (ABS(PXD) .GT. 0.1D0*P3) GOTO 50
     PX = P3 - 2.0D0*PXD
     D4 = P3 - 2.0D0*PXD
     GOTO 55
 50  PX = D4**D3
     D4 = PX
 55  I2 = 1
     GOTO 35
C-----------------------
C    PX .LT. P3
C-----------------------
 60  D5 = 2.0D0 - D3
     IF (I2 .GT. 0) GOTO 100
     IF (NN .GT. 6 .OR. PX .LT. 0.01D0*P3) GOTO 75
     IF (P3 .GT. 0.01D0 .OR. PX .EQ. 0.0D0) GOTO 65
     IF (ABS(PXD) .GT. 0.1D0*P3) GOTO 65
     PX = P3 - 2.0D0*PXD
     D4 = P3 - 2.0D0*PXD
     GOTO 70
 65  PX = D4**D5
     D4 = PX
 70  I2 = -1
     GOTO 35
C---------------------------------------------------------
C    ESTIMATE FOR R USING CIRCUMSCRIBED CUBE. C1 = 1/SQRT(3).
C---------------------------------------------------------
 75  R2 = ELLRC(HX,HY,HZ,SX,SY,SZ,P3,C1*RL,RH,E8,N6)
 80  IF (R2 .LE. RL) GOTO 85
     RP = R2
     R = R2
C-----------------------------------------------
C    COMPUTES ELLCOV(R) AND MAKES PROPER STORAGES
C-----------------------------------------------
     CALL SUB3(P3,R,HX,HY,HZ,SX,SY,SZ,XK0,XK2,S
   +          ,PX,PXD,RL,PL,PLD,RH,PH,PHD,NN,KK)
     IF (ABS(PXD) .LE. E8) RETURN
```

```
 85  R1 = R2/C1
     IF (R1 .LT. RH) RH = R1
     IF ((P3 .LE. 0.2D0) .AND. (R2 .EQ. RP)) GOTO 95
 90  RP = R
     R = 0.5D0*(RL + RH)
     IF (PL .EQ. 0D0 .OR. PH .GT. 1D4*PL) R = (RH + 3*RL)/4
     IF (ABS(RH - RL) .LE. E4*RL) GOTO 190
C-------------------------------------------------
     CALL SUB3(P3,R,HX,HY,HZ,SX,SY,SZ,XK0,XK2,S
   +          ,PX,PXD,RL,PL,PLD,RH,PH,PHD,NN,KK)
 95  IF (ABS(PXD) .LE. E8) RETURN
     IF (ABS(RH - RL) .GT. .1D0*R) GOTO 90
     IF (ABS(R - RP) .GT. 5.D-3*R
   +     .OR. ABS(PH - PL) .GT. 1.D-3*P3) GOTO 105
100  RP = R
     R = RL - (RL - RH)/(PL - PH)*PLD
     GOTO 110
105  RP = R
     R = 0.5D0*(RL + RH)
C-------------------------------------------------
110  CALL SUB3(P3,R,HX,HY,HZ,SX,SY,SZ,XK0,XK2,S
   +          ,PX,PXD,RL,PL,PLD,RH,PH,PHD,NN,KK)
     IF (ABS(PXD).GT. E8) GOTO 115
     R = R - (RL - RH)/(PL - PH)*PXD
     RETURN
115  IF (ABS(RH - RL) .LE. E4*R) GOTO 190
     DO 120 J2 = 1,25
       R = 0.5D0*(RL + RH)
       IF (PL .EQ. 0D0 .OR. PH .GT. 1D4*P3) R = (RH + 3*RL)/4
       CALL SUB3(P3,R,HX,HY,HZ,SX,SY,SZ,XK0,XK2,S
   +          ,PX,PXD,RL,PL,PLD,RH,PH,PHD,NN,KK)
       IF (ABS(PXD).LT. E8) RETURN
       IF (ABS(RH - RL) .LE. E4*R) GOTO 190
       IF (PL .GT. 1D-3*PH) GOTO 125
120    IF (NN .GT. 30) GOTO 185
C-------------------------------------------------------
125  IK = 0
     IF (PL .EQ. 0D0) CALL SUB3(P3,RL,HX,HY,HZ,SX,SY,SZ,XK0,XK2,S
   +          ,PX,PXD,RL,PL,PLD,RH,PH,PHD,NN,KK)
     IF (PH .EQ. 1D0) CALL SUB3(P3,RH,HX,HY,HZ,SX,SY,SZ,XK0,XK2,S
   +          ,PX,PXD,RL,PL,PLD,RH,PH,PHD,NN,KK)
     IF (P3 .GT. .5D0) GOTO 130
C-------------------
C     P3 .LE. .5
C-------------------
     X1 = RL
     X0 = RH
     F0 = PHD
```

```
      F1 = PLD
      GOTO 135
C---------------------
C        P3 .GT. .5
C---------------------
 130  X0 = RL
      X1 = RH
      F0 = PLD
      F1 = PHD
 135  IF (ABS(RH - RL) .GT. 5.D-3*R .OR. ABS(PH - PL) .GT. 1D-3*P3)
     *   GOTO 145
 140  R = 0.5D0*(RL + RH)
      GOTO 150
C----------------------------
C      MODIFIED KING'S PROCEDURE
C----------------------------
 145  R = X1 - F1*(X1 - X0)/(F1 - F0)
      IF (IK .NE. 0) RETURN
 150  CALL SUB3(P3,R,HX,HY,HZ,SX,SY,SZ,XK0,XK2,S
     +          ,PX,PXD,RL,PL,PLD,RH,PH,PHD,NN,KK)
      IF (KK .NE. 0) GOTO 195
      IF (ABS(PXD) .LE. E8) IK = 1
      IF (RH - RL .LT. E4*R) GOTO 190
      IF (NN .GE. 30) GOTO 185
      IF (ABS(PXD - PXDO) .GT. E8) GOTO 155
      IF (IK .NE. 0) GOTO 155
      IF (ABS(PXD) .GT. 1D2*E8) GOTO 155
      RETURN
 155  PXDO = PXD
      IF (PXD*F1 .GT. 0.D0) GOTO 160
      D4 = X1
      X1 = X0
      X0 = D4
      D4 = F1
      F1 = F0
      F0 = D4
C----------------------------
 160  D3 = F1/(F1 + PXD)
      F0 = F0*D3
      X1 = R
      F1 = PXD
 165  R = X1 - F1*(X1 - X0)/(F1 - F0)
      IF (IK .NE. 0) RETURN
      CALL SUB3(P3,R,HX,HY,HZ,SX,SY,SZ,XK0,XK2,S
     +          ,PX,PXD,RL,PL,PLD,RH,PH,PHD,NN,KK)
      IF (KK .NE. 0) GOTO 195
      IF (ABS(PXD) .LE. E8) IK = 1
      IF (RH - RL .LT. E4*R) GOTO 190
```

```
      IF (NN .GE. 30) GOTO 185
      IF (ABS(PXD - PXDO) .GT. E8) GOTO 170
      IF (IK .NE. 0) GOTO 170
      IF (ABS(PXD) .GT. 1D2*E8) GOTO 170
      RETURN
  170 PXDO = PXD
      IF (PXD*F1 .GT. 0.D0) GOTO 160
      X0 = X1
      F0 = F1
      F1 = PXD
      X1 = R
      GOTO 145
C------------
C      EXITS
C-----------------------------------------------
C   P3 .LT. MAX(1D-40,1D6*DPMPAR(2)), IND = -1, R = -1E10.
C-----------------------------------------------
  175 IND = -1
      RETURN
C-----------------------------------------------
C   P3 .GT. 1 - MAX(1D-12,10*DPMPAR(1)), IND = 1, R = -1E10.
C-----------------------------------------------
  180 IND = 1
      RETURN
C-------------------
C      NN .EQ. 30
C-------------------
  185 IND = 2
      RETURN
C--------------------
C    RH - RL .LT. E4*R
C--------------------
  190 IND = 3
      RETURN
C-----------------------
C ACCURACY LIMITED IN ELLCOV
C-----------------------
  195 IND = 4
      RETURN
C-------------
      END
```

```
      FUNCTION ELLRC(H,HK,HL,S1,S2,S3,P3,RMIN,RMAX,E8,N6)
C-----------------------------------------------------------------
C      2*ELLRC = LENGTH OF A SIDE OF A CUBE CENTERED AT (H,HK,HL) THAT
C      CONTAINS THE TRIVARIATE NORMAL PROBABILITY CONTENT P3 WITH (0,0
C      ,0) MEAN AND STANDARD DEVIATIONS (S1,S2,S3). IF N6 = 40, RESULT
C      SUSPECT. RMIN AND RMAX ARE INITIAL UPPER AND LOWER BOUNDS FOR
C      ELLRC. EXIT MADE WHEN LATEST ITERATE FOR ELLRC, F, SATISFIES
C      ABS(F - P3) .LT. E8*P3. E8 SET IN ELINV3.
C-----------------------------------------------------------------
      EXTERNAL              AERF  ,DEPSLN   ,DPMPAR   ,FCN1
C-----------------------------------------------------------------
      DOUBLE PRECISION      AERF  ,DEPSLN   ,DPMPAR   ,ELLRC
      DOUBLE PRECISION      A(3)  ,RA(3) ,T(3)  ,U(3)
C-----------------------------------------------------------------
      DOUBLE PRECISION      B1    ,C     ,E     ,EPS   ,E1    ,E2
      DOUBLE PRECISION      E8    ,F     ,H     ,HK    ,HL    ,P3
      DOUBLE PRECISION      RMAX  ,RMIN  ,R0    ,R1    ,SQ    ,SQPI
      DOUBLE PRECISION      S1    ,S2    ,S3    ,T1    ,T2    ,V1
C-----------------------------------------------------------------
      SQ = 1.414213562373095D0
      SQPI = .5618958354775629D0
C------------------------------------
C         SQPI = 1/SQRT(PI)
C------------------------------------
      N6 = 0
      E = DPMPAR(2)
      E1 = 1D-20
      E1 = E1*E1
      E = MAX(E,E1)
      E2 = -DEPSLN(0)
      EPS = MAX(5D-7,10*DPMPAR(1))
      ELLRC = 1.0D99
      IF (P3 .EQ. 1.0D0) RETURN
      ELLRC = 0.0D0
      IF (P3 .EQ. 0.0D0) RETURN
      U(1) = 1.0D0/(SQ*S1)
      U(2) = 1.0D0/(SQ*S2)
      U(3) = 1.0D0/(SQ*S3)
      A(1) = ABS(H)*U(1)
      A(2) = ABS(HK)*U(2)
      A(3) = ABS(HL)*U(3)
      R1 = RMAX
      R0 = RMIN
      CALL FCN1(R1,A,U,P3,E,RA,T,F,R0,R1,IDEL)
      IF (ABS(F - P3) .GT. E8) GOTO 5
      ELLRC = R1
      RETURN
```

```
 5    IF (F .LE. P3) RETURN
      CALL FCN1(R0,A,U,P3,E,RA,T,F,R0,R1,IDEL)
      IF (ABS(F - P3) .GT. E8) GOTO 10
      ELLRC = R0
      RETURN
10    IF (F .GE. P3) RETURN
      N6 = 0
15    ELLRC = 0.5D0*(R1 + R0)
      CALL FCN1(ELLRC,A,U,P3,E,RA,T,F,R0,R1,IDEL)
      N6 = N6 + 1
      IF (N6 .EQ. 40) RETURN
      IF (IDEL .NE. 0) GO TO 15
      IF (ABS(F - P3) .LT. 0.1D0*P3) GOTO 30
      GO TO 15
20    ELLRC = 0.5D0*(R1 + R0)
25    CALL FCN1(ELLRC,A,U,P3,E,RA,T,F,R0,R1,IDEL)
      N6 = N6 + 1
      IF (N6 .EQ. 40) RETURN
      IF (IDEL .NE. 0) GO TO 15
30    DO 40 J = 1,3
        T1 = 4*A(J)*RA(J)
        C = 1D0
        IF(T1 .GT. E2) GOTO 35
        C = 1D0 + EXP(-T1)
35      T2 = (A(J) - RA(J))
        RA(J) = EXP(-T2*T2)*C*U(J)
40    CONTINUE
      B1 = SQPI*(RA(1)*T(2)*T(3) + RA(2)*T(1)*T(3)
     *       + RA(3)*T(1)*T(2))
      IF (B1 .LE. 0.0D0) GO TO 15
      V1 = (F - P3)/B1
      ELLRC = ELLRC - V1
      IF ((ELLRC .LT. R0) .OR. (ELLRC .GT. R1)) GO TO 20
      IF (ABS(V1) .GE. 5.0D-5*ELLRC) GOTO 25
      IF (ABS(F - P3) .LT. 1.0D-3*P3) RETURN
      IF (ABS(V1) .LE. EPS*ELLRC) RETURN
      GO TO 25
      END
```

```
      SUBROUTINE FCN1(R,A,U,P3,E,RA,T,F,R0,R1,IDEL)
C---------------------------------------------
C       FCN1 USED IN ELLRC, IT GIVES THE PROBABILITY F OF A SHOT FALL-
C       ING UNDER A TRIVARIATE NORMAL DISTRIBUTION IN A CUBE CENTERED
C       AT (H,HK,HL) WITH SIDES OF LENGTH 2R. THE DISTRIBUTION HAS MEAN
C       (0,0,0) WITH STANDARD DEVIATIONS (S1,S2,S3). IF F .LT. DPMPAR(2),
C        IDEL = -1. IF F .GT. 1, IDEL = 1, OTHERWISE IDEL = 0.
C---------------------------------------------
      EXTERNAL            AERF
C---------------------------------------------
      DOUBLE PRECISION    AERF  ,RA(1) ,A(1)  ,T(1)  ,U(1)
C---------------------------------------------
      DOUBLE PRECISION    E     ,F     ,P3    ,R     ,R0    ,R1    ,V
C---------------------------------------------
      F = 1D0
      IDEL = 0
    5 DO 10 J = 1,3
      RA(J) = R*U(J)
      T(J) = AERF(A(J),RA(J))/2
      F = F*T(J)
      IF (F .LT. E) GOTO 20
   10 CONTINUE
      V = F - P3
      IF (V .GT. 0D0) GOTO 15
      R0 = R
      RETURN
   15 R1 = R
      IF (F .GE. 1D0) GO TO 25
      RETURN
   20 IDEL = -1
      R0 = R
      RETURN
   25 IDEL = 1
      RETURN
      END
```

```
      SUBROUTINE SUB3(P3,R3,HX,HY,HZ,SX,SY,SZ,XK0,XK2,S
     +          ,PX,PXD,RL,PL,PLD,RH,PH,PHD,NN,KK)
C------------------------------------------------
C     SUB3 CALLS ELLCOV. ELLCOV COMPARED WITH P3. PX GIVES VALUE
C     OF ELLCOV. PXD GIVES PX - P3. PL = PX, PLD = PXD AND RL = R3
C     IF PXD  .LT. 0. PH = PX, PHD = PXD AND RH = R3 IF PXD .GT. 0. NN
C     GIVES NUMBER OF ELLCOV CALLS. KK .NE. 0 MEANS THE ACCURACY
C     OF ELLCOV NEEDED IS OUT OF REACH.
C------------------------------------------------
      EXTERNAL            ELLCOV
      DOUBLE PRECISION    ELLCOV
C------------------------------------------------
      DOUBLE PRECISION    HX    ,HY   ,HZ   ,PH   ,PHD  ,PL
      DOUBLE PRECISION    PLD   ,PX   ,PXD  ,P3   ,RH   ,RL
      DOUBLE PRECISION    R3    ,S    ,SX   ,SY   ,SZ   ,XK0
      DOUBLE PRECISION    XK2
C------------------------------------------------
      KK = 0
      PX = ELLCOV(R3,HX,HY,HZ,SX,SY,SZ,XK0,XK2,S)
      IF (PX .GT. 0.0D0) NN = NN + 1
      PXD = PX - P3
      IF (PXD .GT. 0.0D0) GO TO 5
      RL = R3
      PL = PX
      IF (PXD .LT. PLD) KK = 1
      PLD = PXD
      RETURN
    5 RH = R3
      PH = PX
      IF (PXD .GT. PHD) KK = 1
      PHD = PXD
      RETURN
      END
```

DISTRIBUTION

	Copies
ATTN OP 961	1
OP 982	1
OP 982E	1
OP 983	1
OP 987	1
CHIEF OF NAVAL OPERATIONS WASHINGTON DC 20350-2000	
ATTN CODE 032F COMMANDER NAVAL FACILITIES ENGINEERING COMMAND 200 STOVALL ST ALEXANDRIA VA 22332-2300	1
ATTN CODE 411 OFFICE OF NAVAL RESEARCH 800 NORTH QUINCY ST ARLINGTON VA 22217	1
ATTN SEA 05R COMMANDER NAVAL SEA SYSTEMS COMMAND NAVAL SEA SYSTEMS COMMAND HEADQUARTERS WASHINGTON DC 20362	1
ATTN LIBRARY COMMANDING OFFICER FLEET ANALYSIS CENTER NAVAL WEAPONS STATION SEAL BEACH CA 90740-5000	1
ATTN CODE 38 COMMANDER U S NAVAL WEAPONS CENTER CHINA LAKE CA 93555-6001	1
ATTN TECHINAL LIBRARY COMMANDING OFFICER NAVAL ORDNANCE MISSILE TEST STATION WHITE SANDS MISSILE RANGE WHITE SANDS MISSILE RANGE NM 88002-5510	1
ATTN LIBRARY SUPERINTENDENT U S NAVAL OBSERVATORY 34TH AND MASSACHUSETTS AVE NW WASHINGTON DC 20392-5100	1
ATTN ATAA-DDD (PETER SHUGART) U S ARMY TRADOC SYSTEMS ANALYSIS ACTIVITY WHITE SANDS MISSILE RANGE NM 88002	1
PRESIDENT NAVAL WAR COLLEGE NEWPORT RI 02841	1
ATTN LIBRARY COMMANDER NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA 92152-5000	1
ATTN NAVY LIBRARY NAVAL OCEANOGRAPHIC AND ATMOSPHERIC OFFICE STENNIS SPACE CENTER BAY ST LOUIS MS 39529-5004	1
ATTN CODE 2303 (A B OWENS) COMMANDING OFFICER NAVAL RESEARCH LABORATORY WASHINGTON DC 20375-5000	1
ATTN DUDLEY KNOX LIBRARY SUPERINTENDENT U S NAVAL POSTGRADUATE SCHOOL MONTEREY CA 93943	1
ATTN CODE 185	1
CODE 1805	1
CODE 1120 (E O'NEILL)	1
CODE 1127 (S BAYLIN)	1
CODE 1129 (E WOLFE)	1
CODE 5220 (TECHNICAL LIBRARY)	1
COMMANDER, DAVID TAYLOR RESEARCH CENTER BETHESDA MD 20084	

DISTRIBUTION (continued)

	Copies
ATTN MATH & SCIENCES DIV LIB OFFICE OF NAVAL RESEARCH WASHINGTON DC 20360	1
COMMANDING OFFICER MARINE AVIATION DETACHMENT PACIFIC MISSILE TEST CENTER PT MUGU CA 93042	1
ATTN DRDAR-TSB-S (STINFO) DIRECTOR U S ARMY BALLISTIC RESEARCH LAB ABERDEEN PROVING GROUND MD 21005	1
ATTN PLANNING STAFF JOINT STRATEGIC TARGET OFFUT AIR FORCE BASE ND 68113	1
DEFENSE TECHNICAL INFORMATION CENTER CAMERON STATION ALEXANDRIA VA 22304-6145	1
ATTN EXCHANGE AND GIFT DIVISION LIBRARY OF CONGRESS WASHINGTON DC 20540	1
ATTN LIBRARY DIRECTOR NATIONAL SECURITY AGENCY FORT GEORGE G MEADE MD 20755	1
ATTN LIBRARY DIRECTOR NATIONAL BUREAU OF STANDARDS GAITHERSBURG MD 20760	1
ATTN LIBRARY D RAND CORPORATION 1700 MAIN ST SANTA MONICA CA 90406	1
ATTN 5642 (DR DONALD E AMOS) SANDIA NATIONAL LABORATORIES PO BOX 5800 ALBUQUERQUE NM 87185	1

	Copies
ATTN DONALD BAKER MOORE EXPLORATORY TECHNOLOGY PO BOX KK FAIRFIELD CA 94533	1
ATTN F4A (DR I SUGAI) THE JOHNS HOPKINS UNIVERSITY APPLIED PHYSICS LABORATORY JOHNS HOPKINS ROAD LAUREL MD 20707	1
ATTN JOHN A SIMPSON NORDEN SYSTEMS INC CHIEF OF MISSILE AND PROJECTILE SYSTEMS DESIGN PO BOX 5300 NORWALK CT 06856	1
ATTN M MILLer SHERING PLOUGH CORP 60 ORANGE ST BLOOMFIELD NJ 07003	1
ATTN R F HAUSMAN 0/8449 B538	1
ATTN DR H WEINGARTEN DEPARTMENT 6212 BLDG 104 LOCKHEED MISSILE & SPACE COMPANY P O BOX 504 SUNNYVALE CA 94088	1
ROBERT S DIDONATO 2600 N HAMPDEN CT APT (7J) CHICAGO IL 60614	1
ATTN PROF H GOLDSTEIN COLUMBIA UNIVERSITY ENGTER BLDG 520 W 120TH ST NEW YORK NY 10027	1
ATTN PROF W KRUSKAL UNIVERSITY OF CHICAGO STATISTICS DEPARTMENT 5734 UNIVERSITY AVE CHICAGO IL 60637	1

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1991

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
Significant Digit Computation of the Ellipsoidal Coverage Function and Its Inverse

5. FUNDING NUMBERS

6. AUTHOR(S)
Armido R. Didonato

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Surface Warfare Center (K104D)
Dahlgren, VA 22448-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NAVSWC TR 91-487

9. SPONSORING/MONITORING AGENCY NAME(S) AND

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

An algorithm is given for evaluating the cumulative trivariate normal probability distribution function, also called the ellipsoidal coverage function, over an offset sphere. A Fortran subprogram, ELLCOV, supplies the function to at least six significant digits over a large range of the input parameters when the precision is not restricted by inherent error. Also, for a given value of the function, the radius R of the sphere with given center is found by the subroutine ELINV3 to at least six significant digits. Several new procedures are used. Listings of the transportable Fortran subprograms ELLCOV and ELINV3, with supporting routines, are included.

14. SUBJECT TERMS

15. NUMBER OF PAGES
85

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and its title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

BLOCK 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If Known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2
NTIS - Leave blank

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief ***(Maximum 200 words)*** factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code ***(NTIS only)***

Block 17.-19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of this page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.